

Heuristic Ternary Error-Correcting Output Codes Via Weight Optimization and Layered Clustering-Based Approach

Xiao-Lei Zhang and Ji Wu

Abstract

One important classifier ensemble for multiclass classification problems is the Error-Correcting Output Code (ECOC). It bridges the multiclass problem and the binary-class classifiers by decomposing the multiclass problem to a serial binary-class problems. In this paper, we present a novel Weight Optimization and Layered Clustering-based ECOC (WOLC-ECOC). WOLC-ECOC is a heuristic ternary ECOC. It starts with an arbitrary valid ECOC ensemble and then iterates the following two steps until the training risk converges. The first step is to train a new binary-class classifier that discriminates the most confusing pair of classes by a novel Layered Clustering based ECOC (LC-ECOC) coding method. The second step is to add the new classifier to the ECOC ensemble effectively by a novel Optimized Weighted (OW) decoding algorithm. Technically, LC-ECOC can construct multiple different strong classifiers on a single binary-class problem by employing a layered clustering-based approach, so that the heuristic training process will not be blocked by some difficult binary-class problem. The OW decoding guarantees the non-increase of the training risk after adding the new binary-class classifier, so that the heuristic training process can be easily controlled via the training risk, which enables WOLC-ECOC maintain a small code length. Moreover, the cutting plane algorithm is employed to make the OW decoding available for large scale problems. The experimental comparison with 15 ECOC coding-decoding methods on 14

Manuscript received June 22, 2012; revised February 18, 2013.

This work was supported in part by the China Postdoctoral Science Foundation funded project under Grant 2012M520278 and in part by the National Natural Science Funds of China under Grant 61170197.

All authors are with Multimedia Signal and Intelligent Information Processing Laboratory, Tsinghua National Laboratory for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing, China (e-mail: huoshan6@126.com, wuji_ee@tsinghua.edu.cn).

Digital Object Identifier

UCI datasets and the application to the music genre classification problem demonstrate the effectiveness of WOLC-ECOC.

Index Terms

Error-Correcting Output Code (ECOC), ensemble learning, multiple classifier system, multiclass classification.

I. INTRODUCTION

Over the last decades, classifier ensembles [1]–[6], such as *bootstrap aggregating (bagging)* [7], *boosting* [8], and their variations [9], [10], have been proven to be effective approaches for solving learning problems like classification and regression [11]–[13]. For such tasks, the success of the ensemble methods relies on a good selection of the base learners and a strong *diversity* among the base learners, where the word “diversity” means that when the base learners make predictions on an identical pattern, they are different from each other in terms of errors. As summarized in [1]–[4], there are generally four groups of classifier ensembles for addressing the diversity. They are the methods of 1) manipulating the training examples [7], 2) manipulating the input features [9], 3) manipulating the training parameters [14], and 4) manipulating the output targets [15].

The manipulation on the output targets, which is also known as Error-Correcting Output Code (ECOC) [15], was originally motivated from the information theory for correcting bits caused by noisy communication channels. Given a multiclass problem, the key idea of ECOC is to assign each class a unique codeword. All codewords formulate an ECOC *coding matrix*. Each row of the coding matrix is a *codeword*, while each column defines a bipartition of the classes. Training *dichotomizers* (i.e. binary-class classifiers) on different bipartitions of the classes respectively results in an ECOC ensemble. ECOC has two notable merits. The first merit is that it bridges a relation between the study of multiclass problems and dichotomizers. This is motivated from the facts that 1) solving a multiclass problem directly as a whole is sometimes difficult, and 2) in practice, it is worthy alleviating the problem to a serial binary problems that can be easily learned by some strong and popular dichotomizers, such as AdaBoost [16] and Support Vector Machine (SVM) [17]–[19]. The second merit is that it has an error-correcting potential by proper codeword designs.

ECOC consists of two parts – *coding* and *decoding*. Coding is the design of the codewords that assigns each class a unique codeword. Decoding is the prediction process that matches the predicted codeword of a test pattern with its most similar class codeword.

In respect of coding, the coding techniques can be categorized to two classes. The first class is the *problem-independent* coding design [15], [20], [21]. It tries to design a coding matrix that has a strong error-correcting ability. This class is motivated from the theory of channel codings. However, because solving a multiclass problem with ECOC is more than an error-correcting issue in channel codings, the error-correcting ability in ECOC is not obvious at present. The second class is the *problem-dependent* coding design [22]–[40], which has attracted much attention in recent years. It tries to find an ECOC ensemble that dedicates to a given multiclass problem without considering the error-correcting ability of the coding matrix much. One state-of-the-art problem-dependent coding design is the *ternary* ECOC-Optimizing Node Embedding (ECOC-ONE) [30], [41], where the word “ternary” means that its coding matrix contains three elements $\{-1, 0, 1\}$, see Section II in detail. It starts with the Discriminant ECOC (DECOC) [29] and then adds dichotomizers that discriminate the most confusing pairs of classes into the ECOC ensemble. However, this heuristic training strategy might be blocked by some difficult binary-class problems. Later on, another famous problem-dependent coding design, called ternary subclass-ECOC [31], is proposed to tackle the problem. It utilizes the subclass splitting technique to separate the difficult binary-class problem to several easier binary-class subproblems. However, one drawback of the subclass-ECOC is that it employs a decision-tree for the class-splitting. As we know, the decision tree has an inherent drawback that if an example is misclassified by a *parent node*, it will have no chance to be corrected by any *child nodes*.

Besides the aforementioned two classes of coding, the diversity, which is the most important element of a classifier ensemble, is far from explored yet. As will be shown in Section II, except for the diversity between codewords, other kinds of diversities, such as the diversity between the base dichotomizers, were almost forgotten in the early ECOC studies, which makes all possible dichotomizers (i.e. all bipartitions of classes) limited and the error-correcting ability inapparent. Only recently, it is becoming more and more attractive, such as manipulating the features [42], [43] and manipulating the parameters of the base dichotomizers [37].

In respect of decoding, one state-of-the-art ECOC decoding approach is the Loss Weighted (LW) decoding [44] for ternary ECOCs [45]. But the weight matrix of the LW decoding are assigned empirically according to the accuracies of the dichotomizers. It is known that the relationship between the accuracy of a classifier ensemble and the accuracies of its base classifiers is not straightforward.

In this paper, we propose a simple heuristic ternary ECOC, called Weight Optimization and Layered Clustering based ECOC (WOLC-ECOC), to address the aforementioned problems. The system overview of WOLC-ECOC is shown in Fig. 1. Specifically, WOLC-ECOC begins with an arbitrary *valid* ECOC

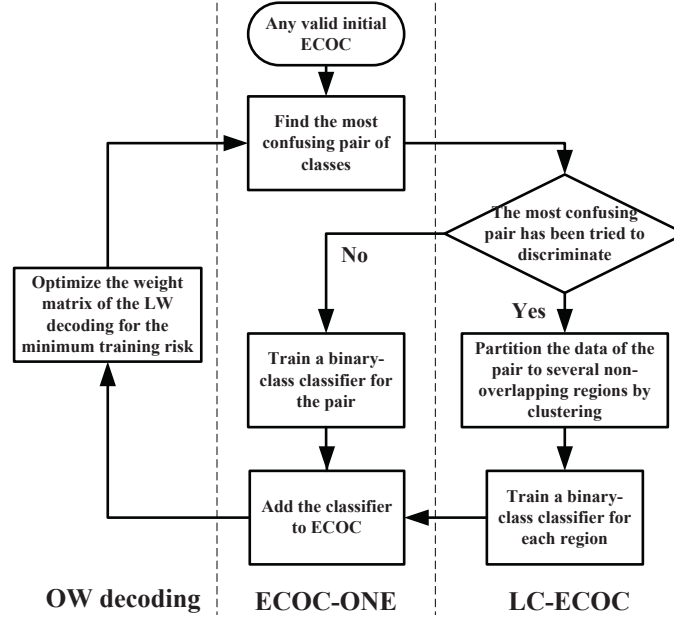


Fig. 1. System overview of the proposed WOLC-ECOC. The building blocks between the dotted lines form the heuristic ECOC-ONE [30], [41]. The building blocks outside the dotted lines are the contributions of this paper.

ensemble and iteratively adds new dichotomizers to the ensemble in a greedy training manner by the following two steps until the training risk converges, where the word “valid” means that the codewords are different from each other. The first step is to train a dichotomizer that discriminates the most confusing pair of classes by a new Layered Clustering-based ECOC (LC-ECOC) approach. The second step is to add the dichotomizer to the ECOC ensemble effectively by a new Optimized Weighted (OW) decoding algorithm. The building blocks between the dotted lines of Fig. 1 form the heuristic ECOC-ONE [30], [41]. The building blocks outside the dotted lines are the contributions of this paper. We summarize the contributions as follows:

- 1) **A novel LC-ECOC coding method is proposed.** It is motivated from the weakness of the famous ECOC-ONE [30], [41] whose heuristic training process might be blocked by some difficult binary-class problems, and from the merit of the subclass-ECOC [31] that splits a difficult problem to several easier sub-problems. The key idea of LC-ECOC is to mine the diversity between the base dichotomizers by manipulating the training examples, which was often forgotten by previous studies. Specifically, it aims to construct multiple different strong dichotomizers on a single pair of classes, where each dichotomizer is trained by first splitting the pair to several different regions via clustering and then training one sub-dichotomizer on each region. Because the regions split from clustering in

different time are different from each other, the base dichotomizers constructed on a single pair is different from each other too.

- 2) **A novel Cutting-Plane Algorithm (CPA) based OW decoding method is proposed.** This new OW decoding method is to improve the LW decoding by optimizing the weight matrix that is only assigned empirically in the LW decoding [44]. Like the LW decoding, the OW decoding method is also a non-biased decoding method for ternary codes, but it can group the base dichotomizers effectively for the minimum training risk. We have presented a similar version of the OW decoding in [46]. The main difference between the two versions is that the objective function in [46] has nP slack variables, each for a possible class of an example, while the objective in this paper consists of only n slack variables, each for an example, where n is the number of the training examples and P denotes the number of the classes. Moreover, we further proposed to solve the optimization problem via Cutting-Plane Algorithm (CPA) [47]–[50]. The CPA based OW decoding has time and storage complexities of both $\mathcal{O}(n)$ which meets the requirement of large scale learning problems.
- 3) **A novel WOLC-ECOC classifier system that integrates LC-ECOC and the OW decoding is proposed.** It is a heuristic ternary ECOC that is inspired from ECOC-ONE [30], [41]. As is shown in Fig. 1, WOLC-ECOC iterates the aforementioned two items until the training risk converges. This integration fuses the merits of the two items together: a) LC-ECOC makes multiple different dichotomizers available for a single binary-class problem, so that the greedy training process will not be blocked by some difficult binary-class problem; b) The OW decoding guarantees the non-increase of the training risk whenever adding a new dichotomizer to the ECOC ensemble, so that the heuristic training can be easily controlled via the training risk, which makes a small code length available.
- 4) **A brief literature survey of ECOC is conducted.** It covers the recent progress of ECOC in respect of both coding and decoding. The relationship between the proposed WOLC-ECOC and the algorithms in literature is also analyzed in the survey.

The experimental comparison with 15 coding-decoding methods on 14 UCI benchmark datasets with 2 kinds of base classifiers shows that the proposed WOLC-ECOC outperforms 15 pairs of coding-decoding methods when using the discrete Adaboost as the base classifier, outperforms 12 pairs of coding-decoding methods when using the Gaussian Radial-Basis-Function (RBF) kernel based SVM as the base classifier, and meanwhile maintains a small code length.

The rest of the paper is organized as follows. In Section II, we conduct a brief literature survey on ECOC and present our motivation. In Section III, we present the LC-ECOC coding method. In Section

IV, we first present the OW decoding method and then employ CPA to further lower its time and storage complexities. In Section V, we first introduce how and why to combine the LC-ECOC coding and the OW decoding into a whole, i.e. WOLC-ECOC, and then present WOLC-ECOC in detail. In Section VI, we report the experimental results on 14 datasets from the UCI machine learning repository, and further apply the proposed algorithm to a real-world problem – music genre classification. Finally, we conclude this paper and present some future work in Section VII.

We first introduce some notations here. Bold small letters, e.g. \mathbf{w} , indicate column vectors. Bold capital letters, e.g. \mathbf{M} and \mathbf{W} , indicate matrices. Letters in calligraphic fonts, e.g. \mathcal{W} , indicate sets, where \mathbb{R}^d denotes a d -dimensional real space. $\mathbf{0}$ ($\mathbf{1}$) is a column vector with all entries being 1 (0).

II. A BRIEF LITERATURE SURVEY OF ECOC

In this section, we take an overview over the coding and decoding strategies of ECOC so as to introduce our motivation.

The error-correcting output codes [15] were inspired from the error-correcting ability of channel coding. It originally views *machine learning as a kind of communication problem in which the identity of the correct output class for a new example is being transmitted over a channel. The channel consists of the input features, the training examples, and the learning algorithm* [15]. Given a P class classification problem with a set of labeled examples $\{(\boldsymbol{\rho}_i, y_i)\}_{i=1}^n$ where $\boldsymbol{\rho}_i \in \mathbb{R}^d$ and $y_i \in \{1, 2, \dots, P\}$ is the label of $\boldsymbol{\rho}_i$, ECOC tries to use Q dichotomizers to address this problem. The relation between the classes and the dichotomizers can be expressed by a *binary* coding matrix $\mathbf{M} \in \{-1, 1\}^{P \times Q}$ or a *ternary* coding matrix $\mathbf{M} \in \{-1, 0, 1\}^{P \times Q}$, where the p -th row of \mathbf{M} expresses the codeword of class p , denoted as \mathbf{c}_p , and the q -th column expresses the q -th dichotomizers, denoted as h_q .

A. Survey on Coding Phase

Two common output codes are the one-versus-all (1vsALL) and one-versus-one (1vs1) matrices [51]. Because the two matrices have no error-correcting ability, some researchers tried to use good channel codes that have large hamming distances between the codewords for correcting the errors introduced by the “channels”. This method is known as problem-independent coding design [15]. However, unlike the channel codes in the communication community, the “channels” in ECOC are influenced by the bipartitions of the classes, which might make the “noise” (errors) of the channels rather high due to improper bipartitions. Furthermore, because there are totally $2^{P-1} - 1$ possible bipartitions in any binary

codes, the code length is limited when the number of the classes P is small [37]. Finally, the error-correcting ability of ECOC is severely limited. Until now, to our best knowledge, few evident proofs showed the error-correcting ability [52]. In most cases, the 1vsALL and 1vs1 codings are still the most powerful ones due to the low level “noise” of their “channels” [53]. Although in [20], [21], Tapia *et al.* declared improved performance with the advanced Low-Density Parity-Check (LDPC) channel codes and a proper arrangement of the classes, there is no analysis on how much LDPC contributes to the performance improvement.

Therefore, the ECOC problem is more properly viewed as the design of a dichotomizer ensemble that bridges powerful dichotomizers and multiclass problems without placing the error-correcting ability in an important consideration. This results in the problem-dependent coding design, which tries to find an ECOC ensemble that dedicates to a given multiclass problem. We review the problem-dependent coding designs as follows:

- 1) **Learning ECOC in a single objective.** In [54], Crammer and Singer tried to find an optimal binary coding matrix in a single objective. Because the discrete binary coding matrix makes the optimization problem *NP-complete*, they further relaxed the binary coding matrix to a *continuous* one, and finally derived the single objective based multiclass-SVM. There are also several similar works that try to learn the optimal coding matrix in a single objective with large margin thoughts [22]–[27]. However, it is worthy noting that the multiclass-SVM does not yield a better performance than the traditional 1vsALL and 1vs1 codings, and even suffers from longer training time [51].

Inspired from the multiclass SVM [54], in [28], Zhong *et al.* further took the optimization of the base dichotomizers into the optimization objective. Because the objective is too complicated, they solved the objective approximately via the Constrained Concave-Convex Procedure (CCCP) [55], [56]. Because CCCP is a non-convex optimization tool, the solution suffers from local minima. Moreover, the continuous coding matrix has to be normalized after each CCCP iteration, which makes the nonmonotonic decrease of the objective value unguaranteed.

Summarizing the aforementioned, it might be difficult and time consuming to learn a problem-dependent coding matrix in a single objective.

- 2) **Ternary coding.** In [45], Allwein *et al.* extended the binary coding matrix to the ternary coding matrix $\mathbf{M} \in \{-1, 0, 1\}^{P \times Q}$. If the entry $M(p, q)$ of the matrix is zero, it indicates that the q -th dichotomizer, denoted as h_q , does not take the training examples with label p into consideration. This method greatly enlarges the number of all possible dichotomizers, and makes each binary-class

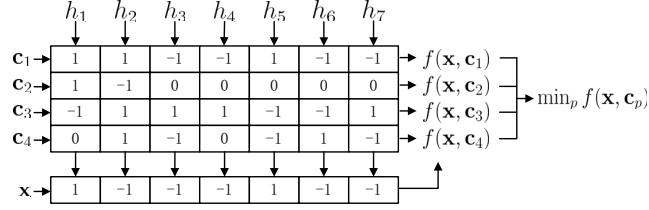


Fig. 2. An example of the ternary ECOC coding matrix \mathbf{M} [44]. In the coding process, if the entry of \mathbf{M} , denoted as $m_{p,q}$, equals to 1, it means that the dichotomizer h_q takes class p as a part of the positive superclass. If $m_{p,q} = -1$, h_q takes class p as a part of the negative superclass. If $m_{p,q} = 0$, h_q does not take the data of class p into training [45]. In the decoding process, taking a test example ρ into h_1, \dots, h_Q successively can get a test codeword of ρ , denoted as $\mathbf{x} = [x_1, \dots, x_Q]^T$. Given a decoding strategy $f(\mathbf{x}, c_p)$, the prediction of ρ can be formulated as a minimization problem $\min_{c_p \in \mathcal{M}} f(\mathbf{x}, c_p)$, where $\mathcal{M} = \{c_p\}_{p=1}^Q$ is the codeword set.

problem focus on a subset of the classes that might be easily differentiated. An example of the ternary ECOC is shown in Fig. 2 with $P = 4$ and $Q = 7$ [44].

In [29], Pujol *et al.* proposed DECOC that embeds a binary decision tree into the ternary code. For each split of the tree, it finds the most discriminative bipartition of the classes at the current tree node, where the discriminability between both sets is maximized in terms of mutual information. It needs at most $P - 1$ dichotomizers. In [57], Yang and Tsang further proposed to find the most discriminative bipartition in terms of maximum separating margin.

However, the intuitive weakness of the decision tree (see Section I) makes DECOC relatively weak in difficult learning problems. To overcome the weakness of the decision tree, later on in [30], [41], Escalera *et al.* and Pujol *et al.* proposed ECOC-ONE that iteratively adds dichotomizers that discriminate the most confusing pairs of classes into the ECOC ensemble, where the most confusing pair is selected according to the *confusion matrix*. Fundamentally, ECOC-ONE is a greedy training algorithm that improves the classification performance directly. It breaks the tree structure of DECOC by embedding new nodes between different branches of the tree, and has shown its power in a broad comparison with other ECOC methods.

However, the performance of ECOC-ONE relies heavily on the discriminability of the base dichotomizer, sometimes, the training process of ECOC-ONE is blocked by its inability on discriminating the most confusing pair. To overcome the block, in [31], Escalera *et al.* further proposed the subclass-ECOC that splits the most confusing class to several subsets, called *subclasses*, by a decision tree. The decision-tree based splitting approach aims to form a serial subclasses that are really easily

learned by the base learner. However, this is not an easy job. Besides the intuitive drawback of the decision tree, it is also hard to decide when to stop splitting. For this, in [31], Escalera *et al.* have used three parameters to control the splitting result. Later on, in [32], Bouzas *et al.* tried to find the optimal parameters by searching the parameter space.

Summarizing the aforementioned, ternary codes are more easily and flexibly trained than both the binary codes and the single objective optimization based methods. Among the ternary codes, the heuristic methods are simple and effective. However, existing heuristic methods might be blocked by some difficult learning problems. The subclass technique has shown the advantage on difficult binary-class problems. However, using decision tree based subclass splitting to ease the confusing class might hinder the subclass-ECOC from practical use. Moreover, if the most confusing classes are still too stubborn to get over after splitting, how to make further improvement? These questions contribute to our motivation on LC-ECOC.

- 3) **Design for diversity enhancement.** Diversity is the base of the classifier systems. All ECOC methods contain somewhat diversities. Here, we focus on the methods that aim at enhancing the diversity of the ECOC ensembles. The following methods focus on manipulating the output codes. In [33]–[35], Kuncheva *et al.* and Escalera *et al.* focused on designing new diversity measures (i.e. measurements of distances) between codewords. In [36], Escalera *et al.* suggested to selectively replace some 0 positions of an original ternary ECOC codes with 1 or -1 according to the accuracies of the base learners at the corresponding classes, which enlarges the diversity between the codewords and leads to better performances than the original ECOC codes. In [38], Escalera *et al.* fused several different DECOC trees for the diversity of the ECOC ensembles. In [58], Hatami tried to delete some columns of a coding matrix that have weak diversities.

However, except for the above methods of manipulating the output codes, other types of diversity have been seldom attempted. Only in [37], Prior and Windeatt manipulated different parameter settings of a special base dichotomizer – Multi-Layer Perceptrons (MLPs) for the diversity of the MLPs. In [42], [43], Bagheri *et al.* manipulated the features by training different base dichotomizers with different feature subsets. Therefore, it might be worth of mining the cross field of different kinds of diversities, which contributes to the generation of our LC-ECOC that manipulates the training examples.

- 4) **Others.** Recently, there are also many other ECOC coding designs and practical applications, such as the expensive evolution computing based methods [39], [40], probability ECOC [59], structured outputs of ECOC [60], online ECOC [61], [62], reject rule based ECOC which rejects the most confusing data [63], [64], etc.

B. Survey on Decoding Phase

The state-of-the-art decoding methods are the Hamming Distance (HD) decoding, euclidean Distance (ED) decoding, probabilistic decoding [65], Loss-Based (LB) decoding [45], and Loss-Weighted (LW) [44] decoding. In this paper, we focus on the LW decoding algorithm since it has a compact theoretical framework and a superior performance to other decoding methods in practice.

In [44], Escalera *et al.* argued that a good decoding strategy should make all codewords (classes) have the same decoding *dynamic range* and zero decoding *dynamic range bias*. Then, they proposed the LW decoding algorithm for the ternary ECOC, which is the unique decoding strategy that satisfy the above decoding condition for the ternary ECOC. The LW decoding introduces a predefined weight matrix $\mathbf{W} = [\mathbf{w}_1^T, \dots, \mathbf{w}_P^T]^T = \begin{bmatrix} w_{1,1} & \dots & w_{1,Q} \\ \vdots & \ddots & \vdots \\ w_{P,1} & \dots & w_{P,Q} \end{bmatrix}$ that has the same size as \mathbf{M} and satisfies the following two constraints

$$w_{p,q} \begin{cases} = 0 & , \text{ if } m_{p,q} = 0 \\ \in [0, 1], & \text{ if } m_{p,q} \neq 0 \end{cases}, \quad \forall p = 1, \dots, P, \forall q = 1, \dots, Q \quad (1)$$

$$\sum_{q=1}^Q w_{p,q} = 1, \quad \forall p = 1, \dots, P \quad (2)$$

where $m_{p,q}$ is an element of the *coding matrix* \mathbf{M} . We denote the set of all feasible weight matrices that are constrained by (2) as \mathcal{W} (i.e. $\mathbf{W} \in \mathcal{W}$). The prediction function of the LW decoding algorithm is given by

$$\min_{\mathbf{c}_p \in \mathcal{M}} f_{LW}(\mathbf{x}, \mathbf{c}_p) = \min_{\mathbf{c}_p \in \mathcal{M}} \sum_{q=1}^Q w_{p,q} \ell(x_q c_{p,q}) \quad (3)$$

where $\ell(\cdot)$ is a user defined loss function. As an example, the linear loss function is defined as $\ell(\theta) = -\theta$.

However, in [44] and its previous works [36], [41], the authors did not mention how to get the optimal \mathbf{W} . Only an empirical assignment is taken from the training accuracy of each base learner. This weakness contributes to the motivation of the proposed OW decoding.

C. Overview of the Literature Survey

To summarize the survey on ECOC, the following items contribute to the motivation of this paper. 1) The design of heuristic ternary codes might be a promising direction. Among the heuristic ternary codes, ECOC-ONE [30] is a good strategy, since it improves the performance in the steepest direction, however, its node embedding process will be blocked when some difficult binary-class problem is encountered.

2) The subclass-ECOC [31] provides a valuable scene that we might split a difficult learning problem to several subproblems for further performance improvement. However, because the method maintains a tree structure as DECOC [29] and needs several tunable parameters to control the subclass splitting, it is difficult to use in practice. 3) The diversity is the cornerstone of classifier ensembles, but currently, most works on ECOC merely focused on constructing diversities between codewords. Only a few works utilized other kinds of diversities [32], [42], [43]. We think that a deep mining on the diversity might be a new growth point. 4) For ternary codes, the LW decoding method is more reasonable than other decoding methods, but its weight assignment is sub-optimal.

In this paper, we focus on inheriting the advantages of ECOC-ONE [30], subclass-ECOC [31] and the LW decoding [44], and meanwhile overcoming their drawbacks.

III. LAYERED CLUSTERING-BASED ECOC

In this section, we will first review the layered clustering-based approach for classifier ensembles [3], and then propose a new layered clustering-based ECOC.

A. Layered Clustering-Based Approach

The layered clustering-based approach [3] was proposed as a classifier ensemble approach. It incorporates the diversity by manipulating the training examples. Specifically, it first splits the training examples to several non-overlapping regions by clustering, where the classification problem in each region is further solved by a classifier. The classifiers in all regions group a super-classifier. Then, it repeats the above procedure several times. Each independent repeat forms a layer of super-classifier. All layers of super-classifiers vote for a test example. This method contains two complementary properties. The first property is that the clustering-based approach can identify overlapping patterns that are hard to differentiate, so that we can get a high accuracy classifier in each layer. But the clustering-based approach do not include any mechanism to incorporate diversity. The second property is that the layered approach uses the mechanism of bagging and boosting to achieve diversities between the super-classifiers for the weakness of the first property. This layered structure, as has been analyzed in [1, page 2], will improve the discriminability of a classifier ensemble on a binary-class problem.

B. LC-ECOC

Inspired by ECOC-ONE [30] and the subclass-ECOC [31], the proposed LC-ECOC also utilizes the greedy training strategy. This strategy tries to iteratively add new dichotomizers that aim at solving

	$h_1^{(s)}$	$h_2^{(s)}$	$h_3^{(s)}$	$h_4^{(c)}$	$h_5^{(c)}$
\mathbf{c}_1	1	0	1	1	0
\mathbf{c}_2	-1	1	-1	-1	1
\mathbf{c}_3	-1	-1	0	0	-1

Initial ECOC

Fig. 3. An example of LC-ECOC for a three-class classification problem. $^{(s)}$ is short for the simple dichotomizer. $^{(c)}$ is short for the clustering-based dichotomizer.

the most difficult binary-class problems to the ECOC ensemble. The main difference between ECOC-ONE, subclass-ECOC and LC-ECOC lies in how they deal with the “*stubborn*” binary-class problem, where the “*stubborn*” problem means that the binary-class problem has already been tried to solve by a dichotomizer in the ECOC ensemble, but it appears to be the most difficult problem again! When encountering this problem, ECOC-ONE has to stop adding new dichotomizers, and the subclass-ECOC employs a decision-tree based subclass splitting technique.

In this paper, LC-ECOC employs the layered clustering-based approach [3] to construct multiple different strong dichotomizers on the stubborn binary-class problem. The key step is that whenever we encounter a stubborn problem, we train one layer of clustering-based dichotomizer on the problem and add it to the ECOC ensemble. Because different layers of the clustering-based dichotomizers on the same stubborn problem can be regarded as different classifiers, LC-ECOC will not be blocked by the stubborn problems.

Fig. 3 gives an example of LC-ECOC for a three-class classification problem. It is initialized with a compact code \mathbf{M} . At the first iteration, it finds the most difficult binary-class problem (or called the most confusing pair of the classes), supposing to be $\mathbf{m} = [1, -1, 0]^T$. Because \mathbf{m} does not exist in \mathbf{M} , LC-ECOC trains a simple base dichotomizer $h_3^{(s)}$ to discriminate class 1 and 2. At the second iteration, when observing the fact that the most difficult problem $[1, -1, 0]^T$ has already appeared as the 3-rd column of \mathbf{M} , it trains one layer of clustering-based dichotomizer $h_4^{(c)}$. So as to $h_5^{(c)}$.

We adopt the heterogeneous clustering-based approach [3], [66] to train each complicated clustering-based dichotomizer. Specifically, in the training process, the heterogeneous clustering-based approach splits the space of the training examples of a pair of classes to N_c regions ($N_c > 1$) without considering the class attributes. For each region, if the region contains examples from both classes, we train a simple base dichotomizer on the region; otherwise, we do nothing to the region. In the prediction process, a test example is first assigned to its *host region*, where the host region is recognized as the one whose center

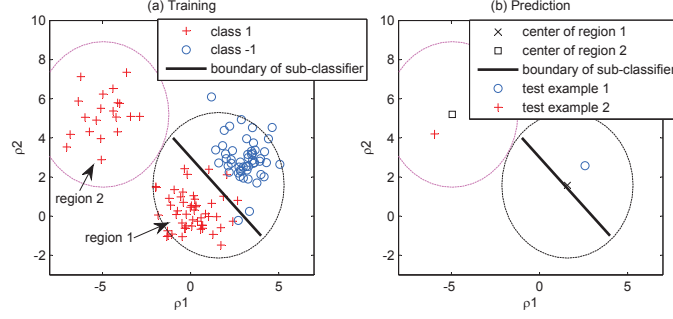


Fig. 4. An example of the heterogeneous clustering-based dichotomizer.

has the minimum Euclidean distance from the example in all regions. Then, if the region owns a base dichotomizer, we predict the test example by the base dichotomizer; otherwise, we set the class attribute of the region to the test example.

Because the main purpose of the clustering-based splitting is to find overlapping patterns and introduce diversities among different layers, the clustering algorithms that have high accuracies, such as spectral clustering [67], agglomerative clustering [68], maximum margin clustering [13], etc., are not suitable for this job. The more “weak” and unstable the clustering algorithm is, the more suitable it seems to be for our job. Hence, the traditional k -means clustering [69] is adopted.

Fig. 4 gives an example of the heterogeneous clustering-based dichotomizer. In Fig. 4 (a), we first find the most confusing region by splitting the training examples to two regions by k -means. Because region 1 consists of two classes, we train a simple base sub-dichotomizer to discriminate the two classes in the region. In Fig. 4 (b), because example 1 falls into region 1, we classify example 1 to class “-1” by the sub-dichotomizer in region 1. Because example 2 falls into region 2 and because region 2 belongs to class “1”, we directly classify example 2 to class “1”.

We summarize LC-ECOC in Algorithm 1.

IV. OPTIMIZED WEIGHTED DECODING FOR ECOC

In this section, we will first propose a new decoding algorithm, named optimized weighted decoding, and then employ the cutting-plane algorithm to accelerate the decoding algorithm for large scale problems.

A. Optimized Weighted Decoding

The OW decoding algorithm is motivated from both the theory of SVM [17]–[19] and the weighted combination of classifiers in the study of classifier ensembles [2], [70], [71]. It tries to optimize the weight

matrix of the LW decoding [44] for the minimal training risk. The optimization problem is formulated as a *linear programming* problem that can be solved in time $\mathcal{O}(n \log n)$. The most similar work in literature is presented in [71].

In the training process, given an example ρ_i with its test codeword being \mathbf{x}_i and its ground truth label being y_i , if ρ_i is classified correctly, according to (3), the following criterion should be satisfied

$$\sum_{q=1}^Q w_{y_i,q} \ell(x_{i,q} c_{y_i,q}) \leq \sum_{q=1}^Q w_{p,q} \ell(x_{i,q} c_{p,q}), \quad \forall p = 1, \dots, P. \quad (4)$$

Let $\mathbf{u}_{i,p} = [\ell(x_{i,1} c_{p,1}), \dots, \ell(x_{i,Q} c_{p,Q})]^T$, Eq. (4) can be rewritten as

$$\mathbf{w}_{y_i}^T \mathbf{u}_{i,y_i} - \mathbf{w}_p^T \mathbf{u}_{i,p} \leq 0, \quad \forall p = 1, \dots, P. \quad (5)$$

If ρ_i is misclassified, it will cause a training loss ξ_i . One possible measurement of ξ_i is the *hinge loss*. It is defined as

$$\xi_i = \max_{p=1,\dots,P} (0, \mathbf{w}_{y_i}^T \mathbf{u}_{i,y_i} - \mathbf{w}_p^T \mathbf{u}_{i,p}). \quad (6)$$

The minimal training risk is achieved by solving the following convex *linear programming* problem

$$\begin{aligned} & \min_{\mathbf{W} \in \mathcal{W}} \mathcal{J}(\mathbf{W}) \\ & \triangleq \min_{\mathbf{W} \in \mathcal{W}} \sum_{i=1}^n \max_{p=1,\dots,P} (0, \mathbf{w}_{y_i}^T \mathbf{u}_{i,y_i} - \mathbf{w}_p^T \mathbf{u}_{i,p}) \end{aligned} \quad (7)$$

which can also be rewritten as the following constrained optimization problem

$$\begin{aligned} & \min_{\mathbf{W} \in \mathcal{W}, \xi_i \geq 0} \sum_{i=1}^n \xi_i \\ & \text{s.t. } \mathbf{w}_p^T \mathbf{u}_{i,p} - \mathbf{w}_{y_i}^T \mathbf{u}_{i,y_i} \geq -\xi_i, \\ & \quad \forall i = 1, \dots, n, \quad \forall p = 1, \dots, P. \end{aligned} \quad (8)$$

Problem (8) can be solved globally in time $\mathcal{O}(n \log n)$. Because this method is inspired by the soft-margin SVM [17]–[19], we also call the parameter ξ_i the slack variable.

If $\mathbf{u}_{i,p}$ only contains discrete values (e.g. setting $\ell(\theta) = -\theta$), the OW decoding algorithm is called the *discrete* OW decoding, otherwise, it is called the *continuous* OW decoding (e.g. setting $\ell(\theta) = \exp(-\theta)$). In this paper, we define $\ell(\theta)$ in (4) as $\ell(\theta) = -\theta$, which means the linear continuous OW decoding is adopted. To prevent unexpected numerical problems, any $\mathbf{u}_{i,p}$ should be normalized to $\mathbf{u}_{i,p}/u^*$, where $u^* = \max_{i,p,q} |u_{i,p,q}|$.

Note that the definition of ξ_i in (6) is very important to the difficulty of the optimization. If the definition is based on the training error directly, i.e. $\xi_i \in \{0, 1\}$, it will cause problem (8) an integer matrix optimization problem that has a time complexity of *NP-complete*. Usually, we use some convex surrogate function, such as hinge loss, to relax ξ_i to a continuous value. As will be shown in Section V-B, this relaxation makes us pick up the most confusing pair of classes according to the *training risk matrix* but not the *confusion matrix*.

B. Cutting-Plane Algorithm Based OW Decoding

Although the OW decoding algorithm has a time complexity of $\mathcal{O}(n \log n)$, we observe that solving problem (8) is still inefficient when the dataset is large-scale (e.g., $n > 10000$). This is caused by the fact that problem (8) has $\mathcal{O}(n)$ parameters and $\mathcal{O}(n)$ constraints. In order to make the OW decoding algorithm available for large scale problems, we employ the well-known CPA [47]–[50], [72] to further lower its time complexity to $\mathcal{O}(n)$. This paper uses the CPA based OW decoding algorithm in all experiments.

CPA is an efficient optimization tool that is good at solving convex optimization problems with large amounts of constraints. Its time and storage complexities are irrelevant to the number of the constraints. In CPA terminology, a problem with a full constraint set is called a master problem [50], while a problem with only a constraint subset from the full set is called a reduced problem, or a cutting-plane subproblem. Generally, CPA begins with a reduced problem that only has an empty working constraint set and then iterates the following two steps:

- 1) Solve the reduced problem with the working constraint set.
- 2) Add the most violated constraint at the current solution point from the full set to the working constraint set, so as to form a new reduced problem.

If the newly generated constraint violates the solution of the reduced problem by no more than ϵ , CPA is stopped, where ϵ is a user defined cutting-plane solution precision. It has been proven that the number of the iterations is upper bounded by $\mathcal{O}(1/\epsilon)$ [49], which is irrelevant to the training set size n .

For our task, we first reformulate problem (8) to the following equivalent optimization problem

$$\begin{aligned}
 & \min_{\mathbf{w} \in \mathcal{W}, \xi \geq 0} \xi \\
 & \text{s.t.} \quad \sum_{i=1}^n \sum_{p=1}^P g_{i,p} (\mathbf{w}_p^T \mathbf{u}_{i,p} - \mathbf{w}_{y_i}^T \mathbf{u}_{i,y_i}) \geq -\xi, \\
 & \quad \forall \mathbf{G} \in \mathcal{Z}^n
 \end{aligned} \tag{9}$$

where $\mathbf{g}_i = [g_{i,1}, \dots, g_{i,P}]^T$, $\mathbf{G} = [\mathbf{g}_1, \dots, \mathbf{g}_n] = \begin{bmatrix} g_{1,1} & \dots & g_{n,1} \\ \vdots & \ddots & \vdots \\ g_{1,P} & \dots & g_{n,P} \end{bmatrix}$, and the set $\mathcal{Z} = \{\mathbf{z}_p\}_{p=1}^P$ with \mathbf{z}_p defined as

$$z_{p,k} = \begin{cases} 1, & \text{if } k = p \\ 0, & \text{otherwise} \end{cases}, \quad k = 1, \dots, P. \quad (10)$$

Problem (8) and problem (9) are equivalent in the following theorem.

Theorem 1: Any solution \mathbf{W} of problem (9) is also a solution of problem (8), and *vice versa*, with $\xi = \frac{1}{n} \sum_{i=1}^n \xi_i$.

Proof: See Appendix A. ■

Comparing problem (9) with (8), we can see that although problem (9) has only 1 slack variable, the number of its constraints is as high as P^n . Fortunately, problem (9) can be solved approximately by CPA with an accurate approximation precision. The CPA based OW decoding algorithm is depicted in Algorithm 2. The derivation is similar with the well-known SVM^{perf} toolbox [48], [72], [73]. We omit the derivation of Algorithm 2.

Because problem (11) has very few constraints, Algorithm 2 has a time complexity of only $\mathcal{O}(n)$ that is consumed on calculating $\sum_{i=1}^n g_{i,p} \mathbf{u}_{i,p}$ in Eq. (11).

Note that the CPA based OW decoding has another significant merit that its storage complexity is irrelevant to the implementation method of the linear programming toolbox. Specifically, the linear programming problem (11) has only $\mathcal{O}(1)$ parameters and $\mathcal{O}(1)$ constraints. We take the standard linear programming toolbox in MATLAB as an example: If we rewrite both Eqs. (8) and (11) to the standard form $\min_{\mathbf{x}} \mathbf{f}^T \mathbf{x}$ s.t. $\mathbf{A} \mathbf{x} \leq \mathbf{b}$, matrix \mathbf{A} in (8) is $(PQ+n) \times n$ in size, while \mathbf{A} in (11) is only $(PQ+1) \times |\Omega|$ in size where $|\Omega|$ is the size of the working constraint set and $|\Omega|$ is a small integer that is irrelevant to n . From this point of view, the original OW decoding cannot handle middle scale datasets in the example environment, while the CPA based OW decoding is not limited by the scale of the dataset.

V. WOLC-ECOC

In this section, we will present a novel weight optimization and layered clustering-based ECOC, which is a combination of LC-ECOC and the OW decoding.

A. Main Idea

The main idea of WOLC-ECOC is to iterates the following two steps until the training risk converges. The first step is to train a dichotomizer for the most confusing pair of classes and add it to the ECOC

ensemble as LC-ECOC does. The second step is to first update the weight matrix of the OW decoding and then find the most confusing pair for the next iteration.

There are two reasons why we update the weight matrix of the OW decoding whenever we add a new dichotomizer to the ECOC ensemble.

The first reason is for the convergence of the training risk. One drawback of the LC-ECOC in Algorithm 1 and other heuristic ECOC algorithms, such as ECOC-ONE [30], is that when a new dichotomizer is added to the ECOC ensemble, there is no guarantee that the training risk will decrease, which means that we do not know when to stop adding new dichotomizers. But if we update the weight matrix of the OW decoding after each time we add a new dichotomizer, the training risk after the addition will not be higher than that before the addition. Formally, given the coding matrix $\mathbf{M}^{(t)}$, WOLC-ECOC classifier ensemble $\mathcal{C}^{(t)}$, minimum training risk $\mathcal{J}_o^{(t)}$, and optimal weight matrix $\mathbf{W}_o^{(t)}$ of the t -th iteration, where $\mathcal{C}^{(t)} = \{h_1, h_2, \dots, h_q\}$ with q being the code length of the t -th iteration, and

$$\mathcal{J}_o^{(t)} = \min_{\mathbf{W}^{(t)} \in \mathcal{W}^{(t)}} \mathcal{J}^{(t)}(\mathbf{W}^{(t)}), \quad (12)$$

$$\mathbf{W}_o^{(t)} = \arg \min_{\mathbf{W}^{(t)} \in \mathcal{W}^{(t)}} \mathcal{J}^{(t)}(\mathbf{W}^{(t)}) \quad (13)$$

with the training risk function $\mathcal{J}^{(t)}(\mathbf{W}^{(t)})$ defined in (7). Supposing that we get a new dichotomizer h_{q+1} at the $t+1$ -th iteration, we can obtain $\mathbf{M}^{(t+1)}$, $\mathcal{C}^{(t+1)}$, $\mathcal{J}_o^{(t+1)}$, and $\mathbf{W}_o^{(t+1)}$ in the same way as we did in the t -th iteration, where $\mathcal{C}^{(t+1)} = \mathcal{C}^{(t)} \cup h_{q+1}$ and $\mathbf{M}^{(t+1)} = [\mathbf{M}^{(t)}, \mathbf{m}]$ with \mathbf{m} denoted as the most difficult binary-class problem in a coding vector form, we have the following theorem:

Theorem 2: The non-increase of the training risk of WOLC-ECOC is guaranteed by the OW decoding, i.e.

$$\mathcal{J}_o^{(0)} \geq \mathcal{J}_o^{(1)} \geq \dots \geq \mathcal{J}_o^{(t)} \geq \mathcal{J}_o^{(t+1)} \geq \dots$$

Proof: See Appendix B. ■

The second reason is for a small ECOC code length. Specifically, discriminating the most difficult binary-class problem at each iteration might help ECOC obtain the maximum performance gain. This is a “gradient-descent” way that makes the training risk converges in a small iteration, so that a small ECOC code length is reachable.

B. Algorithm Description

The framework of the proposed WOLC-ECOC has been presented in Fig. 1 in the introduction section. For its practical use, the blocks have to be instantiated, which is the main content of this subsection.

The training procedure of WOLC-ECOC is described in Algorithm 3. We present the main process of Algorithm 3 as follows.

Initialization. We start with any valid ECOC $\{\mathbf{M}, \mathcal{C}\}$, such as 1vsALL, 1vs1, or compact code based ensembles (i.e. $Q < P$).

Main process. We iterate the following two steps:

- 1) The first step is to optimize the weight matrix \mathbf{W} of the OW decoding and obtain its corresponding minimal training risk \mathcal{J}_o by the *WeightOptimization* function, which is depicted in Section IV.
- 2) The second step is to first find the top s most confusing pairs of classes, denoted as $\{\mathbf{m}_k\}_{k=1}^s$, and then add the s dichotomizers $\{h'_k\}_{k=1}^s$ that discriminate $\{\mathbf{m}_k\}_{k=1}^s$ respectively to the ECOC ensemble. For training h'_k , as presented in LC-ECOC (Algorithm 1), two situations should be considered: If \mathbf{m}_k does not equal to any column of \mathbf{M} , we train a new simple dichotomizer $h'_k{}^{(s)}$ as usual by the *SimpleLearning* function; otherwise, we train a complicated clustering-based dichotomizer $h'_k{}^{(c)}$ by the *ClusteringBasedLearning* function in Section III.

The loop goes on until the maximum iteration number T is reached or the following inequality is satisfied for continuous Z iterations

$$\frac{\mathcal{J}'_o - \mathcal{J}_o}{\mathcal{J}_o} \leq \eta \quad (14)$$

where \mathcal{J}_o and \mathcal{J}'_o are the training risks of the current and previous iterations respectively, and η is a user defined solution precision.

Finally, the ECOC ensemble $\{\mathbf{M}_o, \mathcal{C}_o, \mathbf{W}_o\}$ that achieves the minimum risk is obtained. Here, we have to note that although the OW decoding can reach its global minimum solution at each WOLC-ECOC iteration, the overall heuristic training process can only reach a local minimum solution.

In Algorithm 3, we have considered the following three issues for the robustness and efficiency of WOLC-ECOC.

- 1) **How to balance the discriminability and the code length?** This is a key problem of ECOC. For the proposed method, the layered clustering-based training is a bagging based strategy where multiple layers might trigger a significant performance improvement while one or two layers might contribute to no improvement at all. Therefore, if we stop training immediately when the training risk does not decrease, we may not fully mine the potential of the layered clustering-based approach, but if we construct too many clustering-based dichotomizers, the code length might be too large and there is a risk of overfitting to the training dataset. To solve the problem, we balance the discriminability and the code length in the termination criterion. Specifically, if the training risk does not decrease in a

(a) Confusion matrix

		Predicted class		
		y_1	y_2	y_3
Actual class	y_1	100	0	0
	y_2	0	95	5
	y_3	10	10	80

(b) Training risk matrix

		Predicted class		
		y_1	y_2	y_3
Actual class	y_1	0	0	0
	y_2	0	0	4
	y_3	20	6	0

Fig. 5. A comparison of the confusion matrix and the training risk matrix for a three-class classification problem. (a) Confusion matrix. (b) Training risk matrix.

rate of η (in (14)) for Z continuous iterations, we stop the training procedure. Usually, setting Z to an arrange of $[3 - 5]$ is enough for mining the potential of the layered clustering-based approach.

- 2) **How to define the most confusing pair of classes?** In ECOC-ONE [30], pujol *et al.* have proposed to pick up the most confusing pair of classes according to the confusion matrix. The confusion matrix ϵ is defined as:

$$\epsilon_{i,j} = \sum_{k: \rho_k \in \text{class } i} e_{i,j}(\rho_k) \quad (15)$$

where function $e_{i,j}(\cdot)$ is defined as:

$$e_{i,j}(\rho) = \begin{cases} 1, & \text{if } \rho \in \text{class } i \text{ but is misclassified to } j, \\ 0, & \text{otherwise.} \end{cases}$$

An example of the confusion matrix is shown in Fig. 5 (a). From the figure, we know that 1) each class consists of 100 examples, 2) the candidate confusing pairs of classes are $\mathbf{m}^{1,2} = [1, -1, 0]^T$, $\mathbf{m}^{1,3} = [1, 0, -1]^T$, and $\mathbf{m}^{2,3} = [0, 1, -1]^T$ with the numbers of misclassified examples being $\epsilon_{1,2} = 0 + 0 = 0$, $\epsilon_{1,3} = 10 + 0 = 10$, and $\epsilon_{2,3} = 10 + 5 = 15$ respectively. The most confusing pair is selected as $\mathbf{m}^{2,3}$.

However, in the proposed OW decoding, we have relaxed the range of the integral classification error $\{0, 1\}$ by a convex continuous surrogate function (6) whose range is $[0, +\infty)$. That is to say, what we try to minimize in Algorithm 3 is the training risk $J(\mathbf{W})$ but not the classification error. Hence, the proposed WOLC-ECOC needs to pick up the pair of classes that has the highest training risk

instead of the highest classification error. For this, we propose the training risk matrix ϵ as follows:

$$\epsilon_{i,j} = \sum_{k: \rho_k \in \text{class } i} (\mathbf{w}_i^T \mathbf{u}_{k,i} - \mathbf{w}_j^T \mathbf{u}_{k,j}) \cdot \delta \left(\left(\min_{p=1, \dots, P; p \neq j} \mathbf{w}_p^T \mathbf{u}_{k,p} \right) - \mathbf{w}_j^T \mathbf{u}_{k,j} \right) \quad (16)$$

where $\delta(\cdot)$ is the indicator function defined as follows:

$$\delta(a) = \begin{cases} 1, & \text{if } a > 0, \\ 0, & \text{otherwise.} \end{cases}$$

The training risk matrix records the misclassification risks between classes. Fig. 5 (b) gives an example of the training risk matrix. The training risks of the pairs are $\epsilon_{1,2} = 0 + 0 = 0$, $\epsilon_{1,3} = 0 + 20 = 20$, and $\epsilon_{2,3} = 4 + 6 = 10$ respectively. From the figure, we know that the highest training risk pair is $\mathbf{m}^{1,3}$. Comparing Fig. 5 (a) and Fig. 5 (b), we can see that different optimization objectives might result in different training priorities of a given binary-class problem.

- 3) **How to make the performance robust?** If the most confusing pair is too stubborn to overcome, we may achieve further performance improvement by discriminating the problems that are not so tough. To implement the idea, we discriminate the top s highest training risk pairs of classes, denoted as $\{\mathbf{m}_k\}_{k=1}^s$, but not a single highest training risk pair. This strategy enhances the robustness of WOLC-ECOC.

VI. EXPERIMENTAL ANALYSIS

In this section, we will illustrate the effectiveness and efficiency of the proposed WOLC-ECOC algorithm. Specifically, we will first compare WOLC-ECOC with 15 coding-decoding pairs on 14 UCI benchmark datasets with 2 kinds of base dichotomizers – AdaBoost and SVM. Then, we will study the convergence behavior of WOLC-ECOC. At last, we will apply WOLC-ECOC to the music genre classification problem.

A. Datasets

The data used for experiments consist of 14 multiclass datasets from the UCI Machine Learning Repository database¹. The UCI database is a very large open database for evaluating various machine learning problems, including classification, clustering, regression, information retrieval, etc. Every dataset

¹<http://archive.ics.uci.edu/ml/>

TABLE I
 DESCRIPTIONS OF THE DATASETS. “ n ” IS THE DATASET SIZE, “ d ” IS THE DIMENSION, “ P ” IS THE NUMBER OF THE CLASSES.

ID	Data	n	d	P
1	Dermatology	366	34	6
2	Iris	150	4	3
3	Ecoli	336	7	8
4	Wine	178	13	3
5	Glass	214	9	7
6	Thyroid	215	5	3
7	Vowel	990	10	11
8	Balance	625	4	3
9	Yeast	1484	8	10
10	Satimage	6435	36	7
11	Pendigits	10992	16	10
12	Segmentation	2310	19	7
13	OptDigits	5620	64	10
14	Vehicle	846	18	4

in the database is extracted from a real-world application. The properties of the 14 UCI datasets are listed in Table I. The examples of all datasets are normalized into the range of [0,1] in dimension [74].

B. Experimental Settings

For the proposed WOLC-ECOC, the number of search paths s is set to 3. The termination condition Z is set to 3. The solution precision η is set to 0.01. The initial ECOC is 1vsALL. The maximum iteration number T is set to $3P$ where P is the number of the classes.

To show the effectiveness of WOLC-ECOC, we compare it with 5 state-of-the-art ECOC coding designs, including 1vs1, 1vsALL, random ECOC [45], ECOC-ONE [41] that is initialized by 1vsALL, and DECOC [29]. Each of the competitive coding methods combines with 3 decoding methods, including

HD decoding, LB decoding [45], and LW decoding [44]. We follow the ECOC library [75]² for the implementations of the referenced methods.

To demonstrate how the base classifier affects the performance, two popular base classifiers are chosen. The first one is the discrete AdaBoost [16]. It contains 40 weak learners per strong dichotomizer. The *decision stump* is used as the weak learner. The second one is the Gaussian RBF kernel based SVM. It uses SVM^{perf} [72]³ as the implementation. As was argued by Rifkin and Klautau [53] that the parameters of SVM should be well tuned, we search the parameters of SVM in grid and report the best result. The regularization constant C is searched through $\{2^{12}, 2^{13}, \dots, 2^{18}\}$. The kernel width σ of the RBF kernel is searched through $\{0.25\gamma, 0.5\gamma, \gamma, 2\gamma, 4\gamma\}$, where γ is the average Euclidean distance between examples.

For each dataset, we run each pair of the coding-decoding methods 10 times and record the average experimental results. For each single run, we apply a *stratified sampling* and *ten-fold cross-validation*, and test for confidence interval at 95% with the two-tailed t test. Therefore, we conduct 100 independent runs on each dataset for each pair of coding-decoding methods. Because the experiment runs with 16 pairs of coding-decoding methods and 2 kinds of well-tuned dichotomizers on 14 UCI datasets, we need to conduct 44800 independent runs in total, which is a large-scale experiment that is sufficient to evaluate the effectiveness and efficiency of WOLC-ECOC.

C. Effectiveness

The classification accuracies of different coding-decoding methods with respect to AdaBoost and SVM are listed in Table II and Table III, respectively. From Table II, it is clear that the proposed ECOC method is the most effective one with Adaboost as the base dichotomizer. However, from Table III, we can see that WOLC-ECOC is weaker than the 1vs1 coding method but better than other coding methods when the RBF kernel based SVM is used as the base learner.

The reason why the performance of WOLC-ECOC performs the best with AdaBoost but not the best with the RBF kernel based SVM can be explained in the theory of channel coding. Specifically, it is well known in information theory that the error-correcting ability of any coding method is upper-bounded by the *channel capacity* that is irrelevant to the coding method. It is possible that the performance of a strong coding method in a noisy channel is worse than that of a weak coding method in a clean channel.

²<http://sourceforge.net/projects/ecoclib/>

³The original code can be downloaded from “http://svmlight.joachims.org/svm_perf.html”. The SVM^{perf} in use is a MATLAB version implemented by ourselves.

For the ECOC problem, on one side, as has been presented in the beginning of Section II, the channel is determined by the features, base learner and coding method. The stronger the features and base learner are, the more suitable the bipartitions of the classes are, the more clean the channel will be. On the other side, the more diverse the dichotomizers are, the larger the minimum distance among the codewords is, the more effective the codes will be, i.e., the stronger the error-correcting ability of the code will be, where the term “diverse” is also referred to as *independent* in some papers [42], [43].

Based on the above facts, we analyze the results in the view of channel coding qualitatively as follows. In respect of the ECOC coding methods, 1vs1 might be the most suitable bipartition method that introduces the minimum noise to the channel in most datasets, since it do the bipartitions according to the natural distributions of the data only. In respect of the base learner, the discrete AdaBoost introduces more noise to the channel than the RBF kernel based SVM, since it is known that the discrete AdaBoost is a weaker learner than SVM in most cases. When AdaBoost is used as the base learner, the noise level difference between the channels is not influenced solely by the bipartitions of the data which makes the capacities of the channels similar, hence, the performance is determined by the design of the coding method. This explains the advantage of the proposed WOLC-ECOC in Table II. When SVM is used as the base learner, the noise levels of the channels are significantly influenced by the bipartition methods, because the bipartition in 1vs1 is much less noisy than that in other coding methods, the performance is determined by the channel capacity. This explains the weakness of the proposed method in Table III when compared to 1vs1. To analyze the performance and calculate the channel capacity quantitatively is rather challenging and interesting, we leave it as a future work.

We have to note that the proposed WOLC-ECOC is initialized with 1vsALL in all experiments. If we use other initial codes that are better than 1vsALL, better performances are expectable.

From the tables, we can also conclude that the LW decoding [44] is much more powerful than other decoding methods in the referenced coding methods.

D. Efficiency

The efficiency of an ECOC method is determined by its code length. The shorter the code length is, the more efficient the ECOC method will be.

Table IV lists the average code length of different ECOC coding-decoding pairs. From the table, we can see that although the proposed WOLC-ECOC has an average longer code length than 1vsALL, Random ECOC, ECOC-ONE, and DECOC, it has a much shorter code length than 1vs1. Because it is often worthy sacrificing some efficiency for a much better performance, WOLC-ECOC is efficient in general.

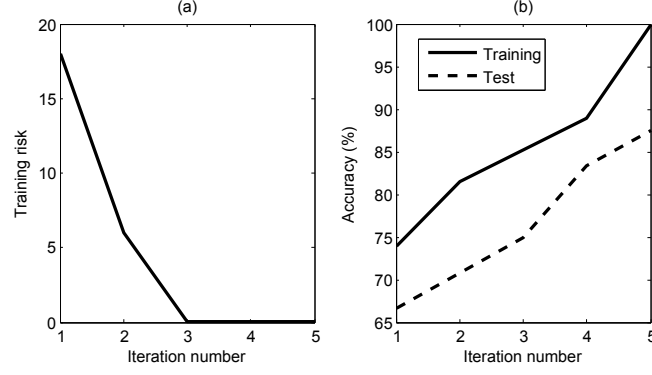


Fig. 6. Convergence behavior of WOLC-ECOC on the Dermatology dataset with discrete AdaBoost as the base learner. (a) Convergence behavior of the training risk (objective value). (b) Curves of the training and test accuracies.

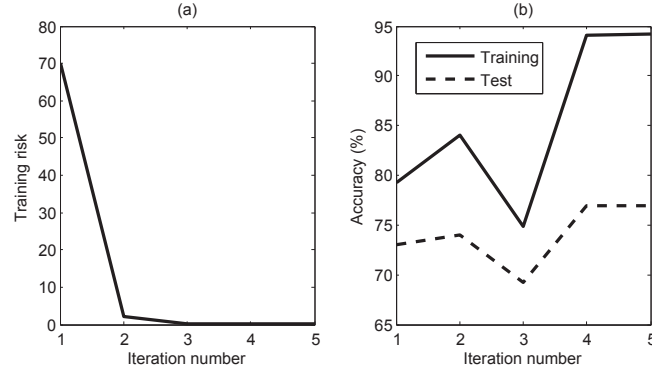


Fig. 7. Convergence behavior of WOLC-ECOC on the Vehicle dataset with discrete AdaBoost as the base learner. (a) Convergence behavior of the training risk (objective value). (b) Curves of the training and test accuracies.

E. Study of the Convergence Behavior

As has been supported by Theorem 2 in Section V-A, the proposed weight optimization method guarantees the non-increase of the objective theoretically. That is to say, the training risk is lowered step by step until convergence. In this subsection, we will verify the convergence behavior of the proposed method empirically. For simplicity, we only give two examples on the Dermatology and Vehicle datasets, which are shown in Figs. 6 and 7 respectively. The same phenomenon can be observed in other datasets.

In both figures, the training risk is defined in Eq. (7), while the accuracy is defined as the ratio of the number of the correctly classified training/test examples over the total number. From the figures, we can see clearly that the training risks (i.e. objective values) decrease rigorously with respect to the

training iteration numbers in both cases. We can also observe that when the objective values decrease, the corresponding training and test accuracies increase in general too. Note that the reason why the accuracy does not increase rigorously with respect to the iteration number is that the training risk we intend to optimize is not equivalent to the classification error (see Section V-B).

F. Application to Music Genre Classification

The developments in information and multimedia technologies enable users to enjoy large amounts of music contents from different media, such as Compact Discs (CDs), the Internet, etc. The large amount of music available calls for developing tools to classify music effectively and efficiently. The music genre classification problem contains two key components – feature extraction and multiclass classifier learning. On one side, there are many kinds of acoustic features, see [76] for an excellent review. On the other side, the SVM based 1vs1 and 1vsALL classifier ensembles are popular for the music classification problems [77].

In this application, we focus on the classifier learning but not on developing new acoustic features. The main purpose of this subsection is to show the advantages of the proposed WOLC-ECOC over the popular 1vs1 and 1vsALL coding methods in the music genre classification problem.

The music genre dataset adopted here is the common Dortmund music dataset [78]⁴. It consists of 1886 recordings of music pieces of 10-s duration. Nine music types are included: *alternative*, *blues*, *electronic*, *folkcountrny*, *funksoulrnb*, *jazz*, *pop*, *raphiphop*, *rock*. The music file format is 44.1kHz, 16-bits, stereo MP3 files. In this study, we first convert each stereo MP3 file to a mono audio file, and then extract three kinds of acoustic features from each audio file exactly as [79] did. They are the Modulation spectral analysis of the Mel-Frequency Cepstral Coefficients (MMFCC), Octave-based Spectral Contrast (MOSC), and Normalized Audio Spectral Envelope (MNASE). Finally, each music file can be seen as an example with three different feature extractions. The parameters of the ECOC methods and the base SVM classifiers are set in the same way as those in Section VI-B.

Tables V and VI list the accuracy and code length comparisons of different coding-decoding methods with the three acoustic features. From Table V, it is clear that the proposed algorithm is the most effective method. It is worthy noting that the proposed algorithm is even more effective than 1vs1. From Table VI, we can see that although the code length of the proposed method is longer than 1vsALL, DECOC,

⁴<http://www-ai.cs.uni-dortmund.de/audio.html>

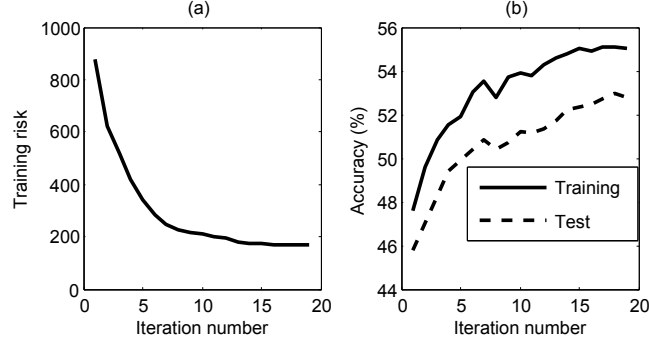


Fig. 8. Convergence behavior of WOLC-ECOC on the Dortmund music genre dataset with MNASE as the feature.

and ECOC-ONE, it is much shorter than 1vs1, which is consistent with the experimental phenomenon on the UCI datasets.

Fig. 8 gives an example of the convergence behavior of the training risk of proposed method with MNASE as the feature. From Fig. 8 (a), we can see that the training risk decreases rigorously with respect to the iteration numbers.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a heuristic ternary weight optimization and layered clustering based error-correcting output code. Specifically, we have first proposed LC-ECOC. LC-ECOC emphasizes the diversity between the base classifiers by employing the layered clustering-based approach, so that it can construct multiple different strong dichotomizers on a given binary-class problem. It helps the heuristic ECOC overcome the block of learning very difficult binary-class problems. Then, we have proposed the CPA based OW decoding. The OW decoding improves the LW decoding by optimizing the weight matrix of the LW decoding for the minimum training risk. The optimization problem is further solved by CPA. The time and storage complexities of the CPA based OW decoding are both linear which meets the requirement of large-scale problems. At last, we have proposed WOLC-ECOC. WOLC-ECOC iteratively executes LC-ECOC and the CPA based OW decoding until the training risk converges. It inherits all merits of LC-ECOC and the CPA based OW decoding. Besides this, because WOLC-ECOC updates the weight matrix after each time we add new classifiers to the ECOC ensemble, its training risk decreases rigorously in the steepest decreasing direction, so that the heuristic training process can be well controlled and a small code length is available.

We have conducted an extensive experimental comparison with 15 state-of-the-art ECOC coding-

decoding pairs on 14 UCI datasets with the discrete AdaBoost and the well-tuned RBF kernel based SVM as two base learners. Experimental results have shown that 1) when Adaboost is used as the base learner, WOLC-ECOC outperforms all referenced coding-decoding pairs; 2) when SVM is used as the base learner, WOLC-ECOC is weaker than the traditional 1vs1 coding method but better than other referenced coding-decoding pairs; 3) WOLC-ECOC has a code length that is much shorter than 1vs1 and comparable with 1vsALL, random ECOC, ECOC-ONE, and DECOC. We have explained the experimental phenomenon in the view of channel coding. We have also applied WOLC-ECOC to the music genre classification problem with SVM as the base learner. Experimental results have shown that WOLC-ECOC outperforms all referenced coding methods including 1vs1 with a relatively short code length.

In the future, we will try to analyze the “channel capacity” of ECOC theoretically so as to guide our ECOC design. We will also try to develop better bipartition techniques on the classes for cleaner “channels” and try to incorporate other types of diversity enhancement techniques so as to make the error-correcting ability of ECOC more apparent.

APPENDIX

A. Proof of Theorem 1

The proof is similar with the proof of [48, Theorem 1]. The key point is to prove that the training loss of problem (9) and the training loss of problem (8) are equivalent:

$$\begin{aligned} \sum_{i=1}^n \xi_i &= \sum_{i=1}^n \max_{p=1, \dots, P} (0, \mathbf{w}_{y_i}^T \mathbf{u}_{i, y_i} - \mathbf{w}_p^T \mathbf{u}_{i, p}) \\ &= \sum_{i=1}^n \max_{\forall \mathbf{g}_i \in \mathcal{Z}} \left(\sum_{p=1}^P g_{i, p} (\mathbf{w}_{y_i}^T \mathbf{u}_{i, y_i} - \mathbf{w}_p^T \mathbf{u}_{i, p}) \right) \end{aligned} \quad (17)$$

where set \mathcal{Z} is defined as $\mathcal{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_P\}$ with \mathbf{z}_p defined as

$$z_{p, k} = \begin{cases} 1, & \text{if } k = p \\ 0, & \text{otherwise} \end{cases}, \quad k = 1, \dots, P. \quad (18)$$

Equation (17) can be reformulated as

$$\sum_{i=1}^n \xi_i = \max_{\forall \mathbf{G} \in \mathcal{Z}^n} \left(\sum_{i=1}^n \sum_{p=1}^P g_{i, p} (\mathbf{w}_{y_i}^T \mathbf{u}_{i, y_i} - \mathbf{w}_p^T \mathbf{u}_{i, p}) \right) = \xi \quad (19)$$

where \mathbf{G} is defined as $\mathbf{G} = [\mathbf{g}_1, \dots, \mathbf{g}_n] = \begin{bmatrix} g_{1,1} & \dots & g_{n,1} \\ \vdots & \ddots & \vdots \\ g_{1,P} & \dots & g_{n,P} \end{bmatrix}$. Theorem 1 is proved.

B. Proof of Theorem 2

We extend the optimal weight matrix $\mathbf{W}_o^{(t)}$ to an equivalent form $\mathbf{W}^{(t+1)'} = [\mathbf{W}_o^{(t)}, \mathbf{0}_{P \times 1}]$. It is easy to know that $\mathbf{W}^{(t+1)'} \in \mathcal{W}^{(t+1)}$. Substituting $\mathbf{W}^{(t+1)'}$ to (7) can yield an objective value that is equivalent to $\mathcal{J}_o^{(t)}$. Because $\mathbf{W}^{(t+1)'}$ is a point in $\mathcal{W}^{(t+1)}$ and problem (7) is a convex optimization problem with $\mathcal{J}_o^{(t+1)}$ as the minimum value, the inequality $\mathcal{J}_o^{(t)} \geq \mathcal{J}_o^{(t+1)}$ holds. Theorem 2 is proved.

ACKNOWLEDGMENT

The authors would like to thank the editors and the anonymous referees for their valuable advice, which greatly improved the quality of this paper. The authors would also like to thank the researchers who opened the codes of their excellent works, which greatly alleviated our workload on constructing the baseline.

REFERENCES

- [1] T. Dietterich, "Ensemble methods in machine learning," in *Proc. Multiple Classifier Syst.* Springer, 2000, pp. 1–15.
- [2] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits Syst. Mag.*, vol. 6, no. 3, pp. 21–45, 2006.
- [3] A. Rahman and B. Verma, "Novel layered clustering-based approach for generating ensemble of classifiers," *IEEE Trans. Neural Netw.*, vol. 22, no. 5, pp. 781–792, 2011.
- [4] M. Re and G. Valentini, "Ensemble methods: a review," 2011.
- [5] K. Leung, F. Cheong, and C. Cheong, "Generating compact classifier systems using a simple artificial immune system," *IEEE Trans. Syst, Man, Cybern, B: Cybern.*, vol. 37, no. 5, pp. 1344–1356, 2007.
- [6] Y. Xu, X. Cao, and H. Qiao, "An efficient tree classifier ensemble-based approach for pedestrian detection," *IEEE Trans. Syst, Man, Cybern, B: Cybern.*, vol. 41, no. 1, pp. 107–117, 2011.
- [7] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
- [8] R. E. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, 1990.
- [9] K. Cherkauer, "Human expert-level performance on a scientific image analysis task by a system using combined artificial neural networks," in *Working Notes of the AAAI Workshop on Integrating Multiple Learned Models*, 1996, pp. 15–21.
- [10] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [11] G. B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst, Man, Cybern, B: Cybern.*, vol. 42, no. 2, pp. 513–529, 2012.
- [12] S. Wang and X. Yao, "Multiclass imbalance problems: Analysis and potential solutions," *IEEE Trans. Syst, Man, Cybern, B: Cybern.*, no. 99, pp. 1–12, 2012.
- [13] X. L. Zhang and J. Wu, "Linearithmic time sparse and convex maximum margin clustering," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, no. 99, pp. 1–24, 2012.
- [14] R. Maclin and J. Shavlik, "Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks," in *Proc. Int. J. Conf. Artif. Intell.*, vol. 14, 1995, pp. 524–531.
- [15] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *J. Artif. Intell. Res.*, vol. 2, pp. 263–286, 1995.

- [16] Y. Freund and R. E. Schapire, “Experiments with a new boosting algorithm,” in *Proc. 13th Int. Conf. Mach. Learn.*, 1996, pp. 148–156.
- [17] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
- [18] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer Verlag, 2000.
- [19] B. Schölkopf and A. J. Smola, *Learning With Kernels*. MIT Press, Cambridge, MA, 2002.
- [20] E. Tapia, J. González, A. Hütermann, and J. García, “Beyond boosting: Recursive ecoc learning machines,” in *Proc. Multiple Classifier Syst.* Springer, 2004, pp. 62–71.
- [21] E. Tapia, P. Bulacio, and L. Angelone, “Recursive ecoc classification,” *Pattern Recogn. Lett.*, vol. 31, no. 3, pp. 210–215, 2010.
- [22] G. Fung and O. Mangasarian, “Multicategory proximal support vector machine classifiers,” *Mach. Learn.*, vol. 59, no. 1, pp. 77–97, 2005.
- [23] J. Weston and C. Watkins, “Support vector machines for multi-class pattern recognition,” in *Proc. 7th European Sym. Artif. Neural Netw.*, vol. 4, no. 6, 1999, pp. 219–224.
- [24] Y. Guermur, “Combining discriminant models with new multi-class svms,” *Pattern Anal. & Applications*, vol. 5, no. 2, pp. 168–179, 2002.
- [25] L. Yoonkyung, L. Yi, and W. Grace, “Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data,” *J. American Statist. Association*, vol. 99, no. 465, pp. 67–81, 2004.
- [26] P. Chen, K. Y. Lee, T. J. Lee, Y. J. Lee, and S. Y. Huang, “Multiclass support vector classification via coding and regression,” *Neurocomputing*, vol. 73, no. 7-9, pp. 1501–1512, 2010.
- [27] S. Ghorai, A. Mukherjee, and P. K. Dutta, “Discriminant analysis for fast multiclass data classification through regularized kernel function approximation,” *IEEE Trans. Neural Netw.*, vol. 21, no. 6, pp. 1020–1029, 2010.
- [28] G. Zhong, K. Huang, and C. Liu, “Learning ECOC and dichotomizers jointly from data,” *Neural Information Processing. Theory and Algorithms*, pp. 494–502, 2010.
- [29] O. Pujol, P. Radeva *et al.*, “Discriminant ecoc: A heuristic method for application dependent design of error correcting output codes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1007–1012, 2006.
- [30] O. Pujol, S. Escalera, and P. Radeva, “An incremental node embedding technique for error correcting output codes,” *Pattern Recogn.*, vol. 41, no. 2, pp. 713–725, 2008.
- [31] S. Escalera, D. M. J. Tax, O. Pujol, P. Radeva, and R. P. W. Duin, “Subclass problem-dependent design for error-correcting output codes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1041–1054, 2008.
- [32] D. Bouzas, N. Arvanitopoulos, and A. Tefas, “Optimizing linear discriminant error correcting output codes using particle swarm optimization,” in *Proc. Int. Conf. Artif. Neural Netw. Mach. Learn.* Springer, 2011, pp. 79–86.
- [33] L. I. Kuncheva and C. J. Whitaker, “Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy,” *Mach. Learn.*, vol. 51, no. 2, pp. 181–207, 2003.
- [34] L. I. Kuncheva, “Using diversity measures for generating error-correcting output codes in classifier ensembles,” *Pattern Recogn. Lett.*, vol. 26, no. 1, pp. 83–90, 2005.
- [35] S. Escalera, O. Pujol, and P. Radeva, “Separability of ternary codes for sparse designs of error-correcting output codes,” *Pattern Recogn. Lett.*, vol. 30, no. 3, pp. 285–297, 2009.
- [36] —, “Recoding error-correcting output codes,” in *Proc. Multiple Classifier Syst.* Springer, 2009, pp. 11–21.
- [37] M. Prior and T. Windeatt, “Over-fitting in ensembles of neural network classifiers within ecoc frameworks,” in *Proc. Multiple Classifier Syst.* Springer, 2005, pp. 834–834.

- [38] S. Escalera, O. Pujol, and P. Radeva, “Boosted landmarks of contextual descriptors and forest-ecoc: A novel framework to detect and classify objects in cluttered scenes,” *Pattern Recogn. Lett.*, vol. 28, no. 13, pp. 1759–1768, 2007.
- [39] M. Bautista, S. Escalera, X. Baro, O. Pujol, P. Radeva, and J. Vitria, “Compact design of ecoc for multi-class object categorization,” *Proceedings of the 5th CVCRD’10, Achievements and New Opportunities in Computer Vision*, pp. 54–57, 2010.
- [40] M. Bautista, X. Baro, O. Pujol, P. Radeva, J. Vitria, and S. Escalera, “Compact evolutive design of error-correcting output codes,” in *Supervised and Unsupervised Ensemble methods and applications-European Conference on Machine Learning*, 2010, pp. 119–128.
- [41] S. Escalera, O. Pujol, and P. Radeva, “Ecoc-one: A novel coding and decoding strategy,” in *Proc. 18th Int. Conf. Pattern Recogn.*, vol. 3, 2006, pp. 578–581.
- [42] M. A. Bagheri, G. Montazer, and E. Kabir, “A subspace approach to error correcting output codes,” *Pattern Recogn. Lett.*, vol. 34, no. 2, pp. 176–184, 2012.
- [43] M. A. Bagheri, Q. Gao, and S. Escalera, “Rough set subspace error-correcting output codes,” in *Proc. 12th Int. Conf. Data Min.*, 2012, pp. 822–827.
- [44] S. Escalera, O. Pujol, and P. Radeva, “On the decoding process in ternary error-correcting output codes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 120–134, 2010.
- [45] E. L. Allwein, R. E. Schapire, and Y. Singer, “Reducing multiclass to binary: A unifying approach for margin classifiers,” *J. Mach. Learn. Res.*, vol. 1, pp. 113–141, 2001.
- [46] X. Zhang, J. Wu, Z. Chen, and P. Lv, “Optimized weighted decoding for error-correcting output codes,” in *Proc. 37th IEEE Int. Conf. Acoustics, Speech Signal Process.*, 2012, pp. 2101–2104.
- [47] J. E. Kelley, “The cutting-plane method for solving convex programs,” *J. Soc. Ind. Appl. Math.*, vol. 8, no. 4, pp. 703–712, 1960.
- [48] T. Joachims, “Training linear SVMs in linear time,” in *Proc. 12th ACM Int. Conf. Knowl. Disc. Data Min.*, 2006, pp. 226–235.
- [49] C. H. Teo, A. Smola, S. V. N. Vishwanathan, and Q. V. Le, “A scalable modular convex solver for regularized risk minimization,” in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2007, pp. 727–736.
- [50] V. Franc and S. Sonnenburg, “Optimized cutting plane algorithm for support vector machines,” in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 320–327.
- [51] C. W. Hsu and C. J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, 2002.
- [52] F. Masulli and G. Valentini, “Effectiveness of error correcting output codes in multiclass learning problems,” in *Proc. Multiple Classifier Syst.* Springer, 2000, pp. 107–116.
- [53] R. Rifkin and A. Klautau, “In defense of one-vs-all classification,” *J. Mach. Learn. Res.*, vol. 5, pp. 101–141, 2004.
- [54] K. Crammer and Y. Singer, “On the learnability and design of output codes for multiclass problems,” *Mach. Learn.*, vol. 47, no. 2, pp. 201–233, 2002.
- [55] A. J. Smola, S. V. N. Vishwanathan, and T. Hofmann, “Kernel methods for missing variables,” in *Proc. 10th Int. Workshop Artif. Intell. Statist.*, 2005, pp. 325–332.
- [56] A. L. Yuille and A. Rangarajan, “The concave-convex procedure,” *Neural Comput.*, vol. 15, no. 4, pp. 915–936, 2003.
- [57] J.-B. Yang and I. W. Tsang, “Hierarchical maximum margin learning for multi-class classification,” in *Proc. 27th Conf. Uncertainty Artif. Intell.*, 2011, pp. 753–760.

- [58] N. Hatami, “Thinned-ecoc ensemble based on sequential code shrinking,” *Expert Systems with Applications*, 2011.
- [59] J. D. Zhou, X. D. Wang, and H. Song, “Research on the unbiased probability estimation of error-correcting output coding,” *Pattern Recogn.*, vol. 44, no. 7, pp. 1552–1565, 2011.
- [60] T. Kajdanowicz, M. Wozniak, and P. Kazienko, “Multiple classifier method for structured output prediction based on error correcting output codes,” *Intell. Inform. Database Syst.*, pp. 333–342, 2011.
- [61] S. Escalera, D. Masip, E. Puertas, P. Radeva, and O. Pujol, “Adding classes online in error correcting output codes framework,” in *Proc. 20th Int. Conf. Pattern Recogn.*, 2010, pp. 2945–2948.
- [62] —, “Online error correcting output codes,” *Pattern Recogn. Lett.*, 2010.
- [63] C. Marrocco, P. Simeone, and F. Tortorella, “Embedding reject option in ecoc through ldpc codes,” *Multiple Classifier Syst.*, pp. 333–343, 2007.
- [64] P. Simeone, C. Marrocco, and F. Tortorella, “Design of reject rules for ecoc classification systems,” *Pattern Recogn.*, vol. 45, pp. 863–875, 2012.
- [65] A. Passerini, M. Pontil, and P. Frasconi, “New results on error correcting output codes of kernel machines,” *IEEE Trans. Neural Netw.*, vol. 15, no. 1, pp. 45–54, 2004.
- [66] B. Verma and A. Rahman, “Cluster oriented ensemble classifier: Impact of multi-cluster characterisation on ensemble classifier learning,” *IEEE Trans. Knowl. Data Eng.*, no. 99, pp. 1–1, 2011.
- [67] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2002.
- [68] X. T. Yuan, B. G. Hu, and R. He, “Agglomerative mean-shift clustering,” *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 2, pp. 209–219, 2012.
- [69] J. MacQueen *et al.*, “Some methods for classification and analysis of multivariate observations,” in *Proc. 5th Berkeley Sym. Math. Statist. Prob.*, vol. 1, pp. 281–297.
- [70] N. Ueda, “Optimal linear combination of neural networks for improving classification performance,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 2, pp. 207–215, 2000.
- [71] L. Zhang and W. D. Zhou, “Sparse ensembles using weighted combination methods based on linear programming,” *Pattern Recogn.*, vol. 44, no. 1, pp. 97–106, 2011.
- [72] T. Joachims, T. Finley, and C. N. J. Yu, “Cutting-plane training of structural SVMs,” *Mach. Learn.*, vol. 77, no. 1, pp. 27–59, 2009.
- [73] T. Joachims and C. N. J. Yu, “Sparse kernel SVMs via cutting-plane training,” *Mach. Learn.*, vol. 76, no. 2, pp. 179–193, 2009.
- [74] C. W. Hsu, C. C. Chang, and C. J. Lin, “A practical guide to support vector classification,” [Online]. Available: <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>, Tech. Rep., 2003.
- [75] S. Escalera, O. Pujol, and P. Radeva, “Error-correcting output codes library,” *J. Mach. Learn. Res.*, vol. 11, pp. 661–664, 2010.
- [76] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, “A survey of audio-based music classification and annotation,” *IEEE Trans. Multimedia*, vol. 13, no. 2, pp. 303–319, 2011.
- [77] T. Li and M. Ogihara, “Toward intelligent music information retrieval,” *IEEE Trans. Multimedia*, vol. 8, no. 3, pp. 564–574, 2006.
- [78] H. Homburg, I. Mierswa, B. Möller, K. Morik, and M. Wurst, “A benchmark dataset for audio classification and clustering,” in *Proc. Int. Conf. Music Information Retrieval*, 2005, pp. 528–531.

- [79] C. H. Lee, J. L. Shih, K. M. Yu, and H. S. Lin, "Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features," *IEEE Trans. Multimedia*, vol. 11, no. 4, pp. 670–682, 2009.

Algorithm 1 LC-ECOC.

```

1: /* Training */
2: repeat
3:   Find the most confusing pair of classes
4:   if the pair is not “stubborn” then
5:     Train a simple dichotomizer as ECOC-ONE [30]
6:   else
7:     Split the space of the pair to  $N_c$  regions by clustering
8:     for  $i = 1, \dots, N_c$  do
9:       if the examples in the  $i$ -th region are from both classes then
10:        Train a sub-dichotomizer on the region
11:      else
12:        Remember the class attribute of the region
13:      end if
14:    end for
15:  end if
16:  Add the new dichotomizer to the ECOC ensemble
17: until the training risk converges
18: /* Prediction */
19: for  $q = 1, \dots, Q$  do
20:   if the dichotomizer  $h_q$  is a simple one then
21:     Predict the example by  $h_q$ 
22:   else
23:     Assign the test example to its host region
24:     if the region owns a sub-dichotomizer then
25:       Predict the example by the sub-dichotomizer of the region
26:     else
27:       Assign the class attribute of the region to the example
28:     end if
29:   end if
30: end for
31: Decode the test codeword of the example

```

Algorithm 2 CPA based OW decoding.

Input: Dataset $\mathcal{U} = \{ \{ \mathbf{u}_{i,p} \}_{p=1}^P, y_i \}_{i=1}^n$.

Output: Optimal weight matrix \mathbf{W} .

Initialization: Arbitrary initial weight matrix \mathbf{W} ($\mathbf{W} \in \mathcal{W}$), empty initial working constraint set $\Omega = \{\}$, the size of working constraint set $|\Omega| \leftarrow 0$.

1: **repeat**

2: $|\Omega| \leftarrow |\Omega| + 1$

3: Calculate the most violated constraint $\mathbf{G}_{|\Omega|}$

$$g_{i,p}^{|\Omega|} = \begin{cases} 1, & \text{if } p = \arg \max_p (\mathbf{w}_{y_i}^T \mathbf{u}_{i,y_i} - \mathbf{w}_p^T \mathbf{u}_p) \\ 0, & \text{otherwise} \end{cases}$$

4: Add the most violated constraint $\mathbf{G}_{|\Omega|}$ to Ω :

$$\Omega \leftarrow \Omega \cup \mathbf{G}_{|\Omega|}$$

5: Solve the reduced problem

$$\begin{aligned} & \min_{\mathbf{W} \in \mathcal{W}, \xi \geq 0} \xi \\ & \text{s.t.} \sum_{i=1}^n \sum_{p=1}^P g_{i,p} (\mathbf{w}_p^T \mathbf{u}_{i,p} - \mathbf{w}_{y_i}^T \mathbf{u}_{i,y_i}) \geq -\xi, \\ & \quad \forall \mathbf{G} \in \Omega \end{aligned} \tag{11}$$

6: **until** Ω is unchanged

Algorithm 3 WOLC-ECOC.

Input: Dataset $\mathcal{D} = \{\rho_i, y_i\}_{i=1}^n$, the number of search paths s , maximum iteration number T , solution precision η , the parameter of the termination condition Z .

Output: ECOC coding matrix \mathbf{M}_o and the corresponding classifier ensemble \mathcal{C}_o , optimal weight matrix \mathbf{W}_o .

Initialization: initial ternary ECOC coding matrix $\mathbf{M} \in \{-1, 0, 1\}^{P \times Q}$ and the classifier ensemble $\mathcal{C} = \{h_1, \dots, h_Q\}$ that is learned from \mathbf{M} and \mathcal{D} , $\mathcal{J}'_o \leftarrow \text{inf}$, $z \leftarrow 0$, $t \leftarrow 0$.

```

1: repeat
2:   for  $i = 1, \dots, n$  do
3:     Predict  $\rho_i$  by the LC-ECOC prediction process
4:     Calculate  $\{\mathbf{u}_{i,p}\}_{p=1}^P$  defined in Eq. (4)
5:   end for
6:   /* Optimize weight matrix */
7:    $\{\mathbf{W}, \mathcal{J}_o\} \leftarrow \text{WeightOptimization}(\mathcal{U}, \mathbf{M})$ , where  $\mathcal{U} = \{\{\mathbf{u}_{i,p}\}_{p=1}^P, y_i\}_{i=1}^n$ 
8:   if  $\mathcal{J}_o == 0$  then
9:      $\mathbf{M}_o \leftarrow \mathbf{M}$ ,  $\mathcal{C}_o \leftarrow \mathcal{C}$ ,  $\mathbf{W}_o \leftarrow \mathbf{W}$ 
10:    return
11:  end if
12:  /* Get the most confusing pairs */
13:  Find  $s$  pairs of classes that have the highest training risks  $\{\mathbf{m}_k\}_{k=1}^s$ . Get their corresponding training risks  $\{\epsilon_k\}_{k=1}^s$ 
14:  /* Learn the base dichotomizers from  $\{\mathbf{m}_k\}_{k=1}^s$  */
15:  for  $k = 1, \dots, s$  do
16:    if  $\epsilon_k \neq 0$  then
17:      if  $\mathbf{m}_k$  does not equal to any column of  $\mathbf{M}$  then
18:         $h'_k \leftarrow \text{SimpleLearning}(\mathcal{D}, \mathbf{m}_k)$ 
19:      else
20:         $h'_k \leftarrow \text{ClusteringBasedLearning}(\mathcal{D}, \mathbf{m}_k)$ 
21:      end if
22:       $\mathbf{M} \leftarrow [\mathbf{M}, \mathbf{m}_k]$ ,  $\mathcal{C} \leftarrow \mathcal{C} \cup h'_k$ 
23:    end if
24:  end for
25:  /* Control the termination criterion */
26:  if  $(\mathcal{J}'_o - \mathcal{J}_o) / \mathcal{J}_o \leq \eta$  then
27:     $z \leftarrow z + 1$ 
28:  else

```

TABLE II

ACCURACY COMPARISON (%) OF DIFFERENT ECOC CODING-DECODING METHODS ON THE UCI DATASETS. THE **DISCRETE ADABOOST** IS USED AS THE BASE LEARNER. IN EACH GRID, THE FIRST LINE IS THE ACCURACY AND THE SECOND LINE IS THE STANDARD DEVIATION. “HD” IS SHORT FOR HAMMING DISTANCE DECODING, “LB” IS SHORT FOR LOSS BASED DECODING, “LW” IS SHORT FOR LOSS WEIGHTED DECODING, AND “OW” IS SHORT FOR OPTIMIZED WEIGHTED DECODING. THE ROW “**RANK**” IS THE AVERAGE RANK OVER ALL 14 DATASETS.

Coding	1vs1			1vsALL			Random			ECOC-ONE			DECOC			WOLC-ECOC
Decoding	HD	LB	LW	HD	LB	LW	HD	LB	LW	HD	LB	LW	HD	LB	LW	OW
Dermatology	91.11 (0.00)	91.11 (0.00)	92.18 (0.00)	87.51 (0.00)	87.51 (0.00)	89.44 (0.00)	81.06 (2.29)	80.86 (3.91)	82.47 (2.96)	89.10 (0.43)	89.23 (0.49)	91.86 (0.00)	70.35 (2.11)	71.19 (1.87)	73.16 (1.84)	91.56 (0.22)
Iris	94.64 (0.00)	94.64 (0.00)	94.64 (0.00)	96.73 (0.00)	96.73 (0.00)	96.03 (0.00)	96.03 (0.46)	95.96 (0.61)	95.89 (0.44)	95.34 (0.00)	95.34 (0.00)	95.62 (0.36)	96.03 (0.00)	96.03 (0.00)	96.03 (0.00)	96.03 (0.00)
Ecoli	85.00 (0.00)	85.00 (0.00)	84.75 (0.00)	81.27 (0.00)	81.27 (0.00)	79.99 (0.00)	76.30 (2.35)	76.63 (1.33)	77.52 (1.36)	80.17 (1.18)	80.16 (1.00)	78.84 (0.83)	75.16 (4.19)	72.36 (4.26)	78.47 (2.37)	87.40 (0.82)
Wine	94.31 (0.00)	94.31 (0.00)	94.31 (0.00)	91.44 (0.00)	91.44 (0.00)	91.44 (0.00)	93.27 (0.88)	93.00 (0.92)	93.20 (0.62)	92.05 (0.55)	91.70 (0.63)	91.87 (0.68)	93.87 (0.56)	93.58 (0.24)	93.93 (0.69)	93.69 (0.00)
Glass	67.78 (0.00)	67.78 (0.00)	67.38 (0.00)	57.12 (0.00)	57.12 (0.00)	68.15 (0.00)	61.81 (1.49)	63.14 (3.14)	63.63 (2.50)	60.45 (1.93)	60.85 (2.63)	65.00 (2.20)	58.21 (3.98)	57.25 (4.35)	63.48 (2.55)	67.28 (0.66)
Thyroid	93.45 (0.00)	93.45 (0.00)	93.45 (0.00)	93.95 (0.00)	93.95 (0.00)	93.95 (0.00)	94.57 (0.92)	94.14 (0.93)	94.16 (0.87)	93.95 (0.00)	93.95 (0.00)	93.95 (0.00)	93.78 (0.60)	93.81 (1.05)	93.93 (0.72)	95.45 (0.00)
Vowel	58.74 (0.00)	58.74 (0.00)	58.74 (0.00)	39.80 (0.00)	39.80 (0.00)	45.97 (0.00)	39.58 (2.60)	37.92 (1.67)	40.99 (1.95)	42.60 (1.65)	42.10 (1.11)	46.50 (1.49)	43.24 (2.74)	45.80 (1.99)	45.28 (2.44)	60.61 (0.82)
Balance	86.40 (0.00)	86.40 (0.00)	86.56 (0.00)	87.52 (0.00)	87.52 (0.00)	87.67 (0.00)	86.75 (1.35)	86.74 (1.96)	87.55 (1.53)	77.49 (0.00)	77.49 (0.00)	77.81 (0.00)	76.70 (0.00)	76.70 (0.00)	76.70 (0.00)	88.97 (0.40)
Yeast	53.93 (0.00)	53.93 (0.00)	53.99 (0.00)	39.24 (0.00)	39.24 (0.00)	54.06 (0.00)	45.48 (0.96)	43.82 (1.99)	45.50 (1.51)	44.96 (1.10)	43.61 (0.93)	50.53 (0.81)	45.51 (1.65)	46.94 (2.15)	50.53 (0.99)	56.28 (0.18)
Satimage	86.84 (0.00)	86.84 (0.00)	86.92 (0.00)	82.36 (0.00)	82.36 (0.00)	82.29 (0.00)	84.70 (0.55)	84.47 (0.90)	85.01 (0.34)	83.26 (0.39)	83.25 (0.24)	83.26 (0.21)	77.69 (2.77)	79.08 (3.47)	84.83 (0.61)	85.74 (0.11)
Pendigits	97.16 (0.00)	97.16 (0.00)	97.24 (0.00)	84.88 (0.00)	84.88 (0.00)	86.25 (0.00)	76.46 (1.03)	76.05 (0.90)	77.65 (1.18)	86.13 (0.24)	86.08 (0.44)	87.13 (0.19)	78.37 (1.00)	77.87 (1.00)	78.84 (1.28)	96.70 (0.15)
Segmentation	95.18 (0.00)	95.18 (0.00)	95.31 (0.00)	90.03 (0.00)	90.03 (0.00)	93.06 (0.00)	91.48 (1.02)	91.28 (1.02)	92.43 (0.69)	92.46 (0.00)	92.46 (0.00)	94.20 (0.00)	93.37 (0.00)	93.37 (0.00)	93.37 (0.00)	95.60 (0.18)
OptDigits	95.03 (0.00)	95.03 (0.00)	95.28 (0.00)	83.27 (0.00)	83.27 (0.00)	84.09 (0.00)	71.66 (1.17)	72.80 (2.06)	74.69 (0.94)	85.80 (0.00)	85.80 (0.00)	86.03 (0.00)	75.27 (0.00)	75.27 (0.00)	75.27 (0.00)	95.67 (0.13)
Vehicle	73.40 (0.00)	73.40 (0.00)	73.52 (0.00)	65.12 (0.00)	65.12 (0.00)	72.33 (0.00)	70.39 (1.00)	70.21 (1.30)	73.07 (0.69)	68.16 (0.81)	67.72 (0.27)	72.35 (0.32)	70.88 (1.31)	71.29 (1.02)	74.28 (1.04)	75.41 (0.13)
Rank	3.93 March 13, 2013	4.29	3.64	9.86	10.07	6.43	8.79	9.50	6.86	8.50	9.07	6.86	8.93	9.14	6.86	2.14

DRAFT

TABLE III

ACCURACY COMPARISON (%) OF DIFFERENT ECOC CODING-DECODING METHODS ON THE UCI DATASETS. THE GAUSSIAN RBF KERNEL BASED SVM IS USED AS THE BASE LEARNER. IN EACH GRID, THE FIRST LINE IS THE ACCURACY AND THE SECOND LINE IS THE STANDARD DEVIATION. “HD” IS SHORT FOR HAMMING DISTANCE DECODING, “LB” IS SHORT FOR LOSS BASED DECODING, “LW” IS SHORT FOR LOSS WEIGHTED DECODING, AND “OW” IS SHORT FOR OPTIMIZED WEIGHTED DECODING. THE ROW “**RANK**” IS THE AVERAGE RANK OVER ALL 14 DATASETS.

Coding	1vs1			1vsALL			Random			ECOC-ONE			DECOC			WOLC-ECOC
Decoding	HD	LB	LW	HD	LB	LW	HD	LB	LW	HD	LB	LW	HD	LB	LW	OW
Dermatology	96.93 (0.59)	96.76 (0.35)	96.88 (0.51)	94.87 (0.46)	94.63 (0.53)	95.82 (0.84)	94.82 (1.33)	95.30 (1.00)	95.94 (0.49)	94.73 (2.88)	94.74 (2.51)	95.59 (0.55)	94.76 (0.71)	95.16 (0.78)	95.40 (0.86)	95.17 (0.55)
Iris	96.80 (0.96)	96.66 (0.63)	96.51 (0.60)	95.41 (1.54)	95.00 (0.72)	96.67 (1.08)	97.52 (0.76)	97.30 (0.38)	96.91 (0.67)	96.32 (0.47)	96.19 (1.33)	96.87 (0.76)	96.97 (0.57)	96.86 (0.79)	96.75 (0.79)	96.69 (0.37)
Ecoli	85.07 (0.81)	85.17 (0.60)	84.81 (0.75)	80.52 (1.03)	80.72 (0.79)	82.75 (0.98)	80.93 (2.15)	81.09 (2.12)	82.40 (1.13)	81.59 (1.13)	81.66 (0.73)	83.28 (0.67)	74.59 (5.18)	74.39 (6.04)	82.70 (1.40)	83.49 (0.25)
Wine	96.05 (1.20)	96.16 (0.79)	96.33 (0.85)	96.65 (0.87)	96.15 (0.78)	96.60 (0.89)	97.37 (0.76)	96.93 (0.93)	97.04 (0.58)	97.16 (0.81)	96.64 (0.63)	96.70 (0.70)	96.38 (0.96)	96.77 (0.67)	96.60 (0.99)	95.85 (0.80)
Glass	62.95 (1.79)	63.84 (2.01)	64.01 (3.16)	52.98 (2.61)	52.03 (1.99)	61.27 (1.37)	61.00 (2.24)	62.09 (2.49)	61.57 (2.03)	56.59 (2.03)	56.60 (2.22)	63.06 (2.54)	58.10 (5.02)	56.75 (4.08)	59.91 (2.76)	63.18 (1.80)
Thyroid	96.20 (0.75)	96.14 (0.60)	96.22 (0.81)	95.21 (1.03)	95.45 (0.96)	95.93 (0.62)	96.21 (0.93)	96.23 (0.69)	95.77 (0.67)	94.99 (0.83)	94.64 (1.05)	95.69 (0.63)	94.50 (1.37)	94.51 (1.27)	93.67 (1.02)	95.63 (0.56)
Vowel	67.11 (1.40)	67.81 (1.19)	67.87 (1.84)	34.88 (0.78)	34.67 (1.58)	36.96 (1.36)	31.07 (2.89)	33.17 (1.88)	34.37 (2.05)	37.56 (2.43)	37.55 (1.69)	39.87 (1.47)	43.40 (3.33)	41.04 (2.54)	41.87 (1.62)	70.87 (1.20)
Balance	90.12 (1.07)	88.89 (1.41)	89.48 (0.99)	90.25 (0.88)	90.34 (1.28)	90.28 (0.93)	89.74 (0.70)	89.78 (0.94)	89.66 (0.98)	88.19 (0.49)	87.71 (0.75)	87.35 (1.55)	88.76 (0.91)	88.95 (0.65)	88.74 (0.61)	91.29 (0.92)
Yeast	58.98 (1.10)	58.95 (0.56)	59.35 (0.63)	38.17 (1.38)	38.41 (1.44)	54.73 (0.62)	51.90 (1.02)	50.97 (2.22)	53.19 (1.35)	43.00 (2.26)	43.71 (2.24)	54.98 (1.02)	51.97 (2.35)	51.97 (3.11)	55.14 (1.33)	55.27 (0.57)
Satimage	85.73 (0.19)	85.74 (0.21)	85.81 (0.20)	80.07 (0.18)	79.98 (0.30)	81.05 (0.27)	81.95 (0.45)	81.43 (0.90)	82.07 (0.58)	81.20 (0.62)	81.27 (0.69)	81.49 (0.81)	74.95 (3.47)	75.86 (3.20)	82.48 (0.68)	86.10 (0.27)
Pendigits	99.01 (0.06)	99.01 (0.06)	98.97 (0.06)	91.79 (0.19)	91.69 (0.15)	92.29 (0.17)	85.19 (1.16)	85.20 (0.76)	86.05 (0.74)	92.53 (0.23)	92.60 (0.16)	93.24 (0.31)	88.23 (1.50)	88.43 (1.15)	88.97 (0.83)	98.25 (0.16)
Segmentation	94.86 (0.45)	95.14 (0.47)	95.06 (0.42)	85.20 (0.98)	85.16 (0.76)	89.45 (0.70)	86.41 (1.54)	86.93 (1.72)	87.60 (1.16)	89.21 (0.78)	89.23 (0.66)	91.79 (0.58)	87.03 (0.89)	86.93 (1.08)	86.67 (1.08)	95.12 (0.30)
OptDigits	97.80 (0.09)	97.74 (0.12)	97.79 (0.07)	92.99 (0.13)	92.88 (0.14)	94.39 (0.15)	88.42 (0.71)	88.18 (1.37)	89.35 (0.81)	94.66 (0.16)	94.74 (0.13)	94.79 (0.18)	89.33 (0.23)	89.33 (0.31)	89.23 (0.17)	97.58 (0.11)
Vehicle	79.62 (0.93)	79.66 (0.83)	80.01 (0.64)	69.02 (0.70)	68.53 (1.43)	75.49 (0.82)	75.16 (0.60)	76.28 (1.62)	77.77 (1.32)	71.34 (1.03)	72.12 (1.06)	76.85 (1.50)	74.48 (0.83)	75.12 (1.74)	76.99 (1.33)	82.51 (0.34)
Rank	2.07 March 13, 2013	2.79	2.43	10.64	10.86	6.14	8.29	6.93	6.07	8.36	8.79	5.07	9.71	9.21	7.29	4.57

DRAFT

TABLE IV
CODE LENGTH COMPARISON OF DIFFERENT ECOC METHODS ON THE UCI DATASETS.

Coding	1vs1	1vsALL	Random	ECOC-ONE						DECOC	WOLC-ECOC	
Decoding	–	–	–	HD		LB		LW		–	OW	
Base classifier	–	–	–	Ada	SVM	Ada	SVM	Ada	SVM	–	Ada	SVM
Dermatology	15.00 (0.00)	6.00 (0.00)	10.00 (0.00)	7.09 (0.08)	7.26 (0.37)	7.11 (0.09)	7.30 (0.28)	7.50 (0.00)	7.74 (0.42)	5.00 (0.00)	9.09 (1.27)	6.00 (0.00)
Iris	3.00 (0.00)	3.00 (0.00)	10.00 (0.00)	4.50 (0.00)	4.93 (0.81)	4.50 (0.00)	4.99 (0.55)	6.31 (0.26)	6.68 (0.39)	2.00 (0.00)	7.00 (0.00)	5.93 (0.78)
Ecoli	28.00 (0.00)	8.00 (0.00)	10.00 (0.00)	9.48 (0.13)	9.05 (0.09)	9.46 (0.13)	9.10 (0.13)	9.65 (0.20)	9.29 (0.20)	7.00 (0.00)	14.75 (2.01)	15.24 (4.16)
Wine	3.00 (0.00)	3.00 (0.00)	10.00 (0.00)	7.00 (0.00)	7.00 (0.45)	7.00 (0.00)	7.36 (0.36)	7.00 (0.00)	7.36 (0.38)	2.00 (0.00)	3.00 (0.00)	3.00 (0.00)
Glass	15.00 (0.00)	6.00 (0.00)	10.00 (0.00)	7.35 (0.13)	7.40 (0.15)	7.23 (0.11)	7.39 (0.18)	7.93 (0.41)	7.61 (0.32)	5.00 (0.00)	9.44 (0.37)	12.50 (1.08)
Thyroid	3.00 (0.00)	3.00 (0.00)	10.00 (0.00)	6.63 (0.00)	6.33 (0.32)	6.63 (0.00)	6.45 (0.74)	6.63 (0.00)	6.18 (0.56)	2.00 (0.00)	3.00 (0.00)	3.35 (0.26)
Vowel	55.00 (0.00)	11.00 (0.00)	10.00 (0.00)	12.10 (0.11)	12.20 (0.13)	12.05 (0.06)	12.23 (0.15)	12.10 (0.11)	12.05 (0.06)	10.00 (0.00)	26.64 (0.58)	24.25 (2.59)
Balance	3.00 (0.00)	3.00 (0.00)	10.00 (0.00)	8.00 (0.00)	7.93 (0.16)	8.00 (0.00)	7.69 (0.39)	8.00 (0.00)	7.71 (0.45)	2.00 (0.00)	15.16 (1.96)	13.60 (3.29)
Yeast	45.00 (0.00)	10.00 (0.00)	10.00 (0.00)	11.20 (0.15)	11.13 (0.10)	11.14 (0.12)	11.11 (0.09)	12.73 (0.29)	11.19 (0.24)	9.00 (0.00)	13.30 (0.63)	16.45 (2.55)
Satimage	15.00 (0.00)	6.00 (0.00)	10.00 (0.00)	7.09 (0.10)	7.06 (0.07)	7.04 (0.08)	7.10 (0.13)	7.00 (0.00)	7.60 (0.32)	5.00 (0.00)	10.70 (2.52)	16.78 (5.18)
Pendigits	45.00 (0.00)	10.00 (0.00)	10.00 (0.00)	11.43 (0.17)	11.10 (0.11)	11.38 (0.18)	11.13 (0.17)	11.04 (0.06)	11.09 (0.12)	9.00 (0.00)	24.06 (4.43)	22.74 (6.23)
Segmentation	21.00 (0.00)	7.00 (0.00)	10.00 (0.00)	8.00 (0.00)	8.00 (0.00)	8.00 (0.00)	8.00 (0.00)	8.25 (0.00)	8.08 (0.09)	6.00 (0.00)	13.18 (2.05)	14.71 (3.30)
OptDigits	45.00 (0.00)	10.00 (0.00)	10.00 (0.00)	11.00 (0.00)	11.00 (0.00)	11.00 (0.00)	11.00 (0.00)	11.00 (0.00)	11.08 (0.12)	9.00 (0.00)	22.45 (5.62)	24.14 (1.93)
March 13, 2013 Vehicle	6.00 (0.00)	4.00 (0.00)	10.00 (0.00)	5.05 (0.07)	5.09 (0.12)	5.02 (0.05)	5.00 (0.00)	5.43 (0.43)	5.68 (0.46)	2.00 (0.00)	10.96 (0.68)	12.56 (4.60)

TABLE V

ACCURACY COMPARISON (%) OF DIFFERENT ECOC CODING-DECODING METHODS ON THE DORTMUND MUSIC GENRE DATASET WITH DIFFERENT FEATURES. THE **GAUSSIAN RBF KERNEL BASED SVM** IS USED AS THE BASE LEARNER. IN EACH GRID, THE FIRST LINE IS THE ACCURACY AND THE SECOND LINE DENOTES THE CORRESPONDING DECODING METHOD. “LW” IS SHORT FOR LOSS WEIGHTED DECODING, AND “OW” IS SHORT FOR OPTIMIZED WEIGHTED DECODING.

Coding	1vs1	1vsALL	DECOC	ECOC-ONE	WOLC-ECOC
MMFCC	43.15	47.33	45.34	49.00	50.49
	LW	LW	LW	LW	OW
MOSC	44.41	47.89	46.76	50.15	52.78
	LW	LW	LW	LW	OW
MNASE	45.75	50.85	46.42	50.93	52.86
	LW	LW	LW	LW	OW

TABLE VI

CODE LENGTH COMPARISON OF DIFFERENT ECOC CODING-DECODING METHODS ON THE DORTMUND MUSIC GENRE DATASET WITH DIFFERENT FEATURES.

Coding	1vs1	1vsALL	DECOC	ECOC-ONE	WOLC-ECOC
MMFCC	45.00	9.00	8.00	16.62	27.64
MOSC	45.00	9.00	8.00	14.24	22.78
MNASE	45.00	9.00	8.00	14.75	24.23